CollegeBoard AP

INCLUDES

- Course framework
- Instructional section
- Sample exam questions
- Create performance task guidelines

AP® Computer Science Principles

COURSE AND EXAM DESCRIPTION

Effective
Fall 2023

AP® Computer Science Principles

COURSE AND EXAM DESCRIPTION

Effective
Fall 2023

AP COURSE AND EXAM DESCRIPTIONS ARE UPDATED PERIODICALLY

Please visit AP Central (apcentral.collegeboard.org) to determine whether a more recent course and exam description is available.

What AP® Stands For

Thousands of Advanced Placement teachers have contributed to the principles articulated here. These principles are not new; they are, rather, a reminder of how AP already works in classrooms nationwide. The following principles are designed to ensure that teachers' expertise is respected, required course content is understood, and that students are academically challenged and free to make up their own minds.

1. AP stands for clarity and transparency. Teachers and students deserve clear expectations. The Advanced Placement Program makes public its course frameworks and sample assessments. Confusion about what is permitted in the classroom disrupts teachers and students as they navigate demanding work.
2. AP is an unflinching encounter with evidence. AP courses enable students to develop as independent thinkers and to draw their own conclusions. Evidence and the scientific method are the starting place for conversations in AP courses.
3. AP opposes censorship. AP is animated by a deep respect for the intellectual freedom of teachers and students alike. If a school bans required topics from their AP courses, the AP Program removes the AP designation from that course and its inclusion in the AP Course Ledger provided to colleges and universities. For example, the concepts of evolution are at the heart of college biology, and a course that neglects such concepts does not pass muster as AP Biology.
4. AP opposes indoctrination. AP students are expected to analyze different perspectives from their own, and no points on an AP Exam are awarded for agreement with a specific viewpoint. AP students are not required to feel certain ways about themselves or the course content. AP courses instead develop students' abilities to assess the credibility of sources, draw conclusions, and make up their own minds.

   As the AP English Literature course description states: "AP students are not expected or asked to subscribe to any one specific set of cultural or political values, but are expected to have the maturity to analyze perspectives different from their own and to question the meaning, purpose, or effect of such content within the literary work as a whole."
5. AP courses foster an open-minded approach to the histories and cultures of different peoples. The study of different nationalities, cultures, religions, races, and ethnicities is essential within a variety of academic disciplines. AP courses ground such studies in primary sources so that students can evaluate experiences and evidence for themselves.
6. Every AP student who engages with evidence is listened to and respected. Students are encouraged to evaluate arguments but not one another. AP classrooms respect diversity in backgrounds, experiences, and viewpoints. The perspectives and contributions of the full range of AP students are sought and considered. Respectful debate of ideas is cultivated and protected; personal attacks have no place in AP.
7. AP is a choice for parents and students. Parents and students freely choose to enroll in AP courses. Course descriptions are available online for parents and students to inform their choice. Parents do not define which college-level topics are suitable within AP courses; AP course and exam materials are crafted by committees of professors and other expert educators in each field. AP courses and exams are then further validated by the American Council on Education and studies that confirm the use of AP scores for college credits by thousands of colleges and universities nationwide.

The AP Program encourages educators to review these principles with parents and students so they know what to expect in an AP course. Advanced Placement is always a choice, and it should be an informed one. AP teachers should be given the confidence and clarity that once parents have enrolled their child in an AP course, they have agreed to a classroom experience that embodies these principles.

© 2023 The Advanced Placement Program, AP, AP Central, and the acorn logo are registered trademarks of the College Board. All other products and services may be trademarks of their respective owners. Visit College Board on the web: **collegeboard.org**.

Contents

v **Acknowledgments**

1 **About AP**

4 **AP Resources and Supports**

6 **Instructional Model**

7 **About the AP Computer Science Principles Course**
7 College Course Equivalent
7 Prerequisites

COURSE FRAMEWORK

11 **Introduction**

13 **Course Framework Components**
15 Computational Thinking Practices
17 Course Content

20 **Course at a Glance**

23 **Big Idea Guides**
24 Using the Big Idea Guides
27 **BIG IDEA 1:** Creative Development
41 **BIG IDEA 2:** Data
57 **BIG IDEA 3:** Algorithms and Programming
97 **BIG IDEA 4:** Computer Systems and Networks
109 **BIG IDEA 5:** Impact of Computing

INSTRUCTIONAL APPROACHES

129 **Selecting and Using Course Materials**

132 **Instructional Strategies**

140 **Developing Computational Thinking Practices**

152 **Using Strategies for Collaboration**

153 **Differentiating Computer Science Instruction**

CURRICULUM ALIGNMENT

157 **Curriculum Alignment**

158 **Unit at a Glance**

EXAM INFORMATION

163 **Exam Overview**

175 **Sample Exam Questions**

SCORING GUIDELINES

188 **Video, Program Requirements, and Written Response 1**

191 **Written Response 2**

STUDENT HANDOUTS

197 **Create Performance Task**

203 **Guidelines for Completing the Performance Task**

APPENDIX

211 APPENDIX 1: **AP CSP Exam Reference Sheet**

219 APPENDIX 2: **AP CSP Conceptual Framework**

Acknowledgments

In partnership with the National Science Foundation, the AP Program collaborated with secondary and postsecondary educators and members of computer science education professional organizations to develop the AP Computer Science Principles course framework.

College Board would like to acknowledge the contributors and reviewers for their assistance with and commitment to the development of this course. All individuals' names and their affiliations were current at the time of contribution.

Christine Alvarado, *University of California, San Diego, CA*

Bradley Bearden, *Dadeville High School, Dadeville, AL*

Daniel Bonarigo, *Riverside Brookfield High School, Chicago, IL*

Adam Cannon, *Columbia University, New York, NY*

Joseph Coglianese, *Troy High School, Fullerton, CA*

Jacqueline Corricelli, *West Hartford Public School, West Hartford, CT*

Tom Cortina, *Carnegie Mellon University, Pittsburgh, PA*

Sandy Czajka, *Riverside Brookfield High School, Riverside, IL*

Marilyn Fitzpatrick, *Charles H. Flowers High School, Springdale, MD*

Dan Garcia, *University of California, Berkeley, CA*

Jeff Gray, *The University of Alabama, Tuscaloosa, AL*

Jessica Jarboe, *Milton High School, Milton, MA*

Brande Jones, *Woodward Academy, College Park, GA*

Douglas Kiang, *Punahou High School, Honolulu, HI*

Jennifer Rosato, *The College of St. Scholastica, Duluth, MN*

Alexander Schenk, *Collegiate School of Medicine and Bioscience, St. Louis, MO*

Cheryl Swanier, *University of Massachusetts - Amherst, Amherst, MA*

Paul Tymann, *Rochester Institute of Technology, Rochester, NY*

Chinma Uche, *CREC Academy of Aerospace and Engineering, Windsor, CT*

Jill Westerlund, *Hoover High School, Hoover, AL*

Carol Yarbrough, *Alabama School of Fine Arts, Birmingham, AL*

College Board Staff

Shu-Kang Chen, *Executive Director, AP STEM*

Crystal Furman, *Director, AP Computer Science Principles*

Laura Keegan, *Director, Product Development and Editorial AP Classroom Instruction*

Daniel Klag, *Senior Director, Assessment Specialist, Computer Science*

Dana Kopelman, *Executive Director, AP Assessment and Assessment Production*

Claire Lorenz, *Senior Director, AP Classroom Instruction Products*

Daniel McDonough, *Senior Director, AP Content and Assessment Publications*

Maureen Reyes, *Executive Director, AP Course Innovations*

Monica Roman, *Senior Director, AP Program Management*

Natalya Tabony, *Senior Director, AP Strategy*

© 2023 College Board

Allison Thurber, *Vice President, AP Curriculum and Assessment*
Jason VanBilliard, *Senior Director, AP Math & Computer Science Department Head*

SPECIAL THANKS
Jan Cuny, Owen Astrachan, Amy Briggs, and the National Science Foundation

© 2023 College Board

About AP

College Board's Advanced Placement® Program (AP®) enables willing and academically prepared students to pursue college-level studies—with the opportunity to earn college credit, advanced placement, or both—while still in high school. Through AP courses in 38 subjects, each culminating in a challenging exam, students learn to think critically, construct solid arguments, and see many sides of an issue—skills that prepare them for college and beyond. Taking AP courses demonstrates to college admission officers that students have sought the most challenging curriculum available to them, and research indicates that students who score a 3 or higher on an AP Exam typically experience greater academic success in college and are more likely to earn a college degree than non-AP students. Each AP teacher's syllabus is evaluated and approved by faculty from some of the nation's leading colleges and universities, and AP Exams are developed and scored by college faculty and experienced AP teachers. Most four-year colleges and universities in the United States grant credit, advanced placement, or both on the basis of successful AP Exam scores—more than 3,300 institutions worldwide annually receive AP scores.

AP Course Development

In an ongoing effort to maintain alignment with best practices in college-level learning, AP courses and exams emphasize challenging, research-based curricula aligned with higher education expectations.

Teachers can choose to adopt the curriculum of one of the AP endorsed providers or design their own curriculum for AP Computer Science Principles, selecting appropriate college-level readings, assignments, and resources. This course and exam description presents the content and skills that are the focus of the corresponding college course and that appear on the AP Exam. The intention of this publication is to respect teachers' time and expertise by providing a roadmap that they can modify and adapt to their local priorities and preferences. By organizing the AP course content and skills into topics, the AP Program is able to provide teachers and students with formative Topic Questions that teachers can assign throughout the year to measure students' progress as they acquire content knowledge and develop skills.

Enrolling Students: Equity and Access

College Board strongly encourages educators to make equitable access a guiding principle for their AP programs by giving all willing and academically prepared students the opportunity to participate in AP. We encourage the elimination of barriers that restrict access to AP for students from ethnic, racial, and socioeconomic groups that have been traditionally underserved. College Board also believes that all students should have access to academically challenging coursework before they enroll in AP classes, which can prepare them for AP success. It is only through a commitment to equitable preparation and access that true equity and excellence can be achieved.

Offering AP Courses: The AP Course Audit

The AP Program unequivocally supports the principle that each school implements its own curriculum that will enable students to develop the content understandings and skills described in the course framework.

The AP Program does have a short list of curricular and resource requirements that must be fulfilled before a school can label a course "Advanced Placement" or "AP." Schools wishing to offer AP courses must participate in the AP Course Audit, a process through which AP teachers' course materials are reviewed by college faculty. Teachers can also adopt a syllabus from an AP endorsed provider during this process.The AP Course Audit was created to provide teachers and administrators with clear guidelines on curricular and resource requirements for AP courses and to help colleges and universities validate courses marked "AP" on students' transcripts. This process ensures that AP teachers' courses meet or exceed the curricular and resource expectations that college and secondary school faculty have established for college-level courses.

© 2023 College Board

The AP Course Audit form is submitted by the AP teacher and the school principal (or designated administrator) to confirm awareness and understanding of the curricular and resource requirements. A syllabus or course outline, detailing how course requirements are met, is submitted by the AP teacher for review by college faculty.

Please visit **collegeboard.org/apcourseaudit** for more information to support the preparation and submission of materials for the AP Course Audit.

How the AP Program Is Developed

The scope of content for an AP course and exam is derived from an analysis of syllabi and course offerings of colleges and universities. Using this research and data, a committee of college faculty and expert AP teachers work within the scope of the corresponding college course to articulate what students should know and be able to do upon the completion of the AP course. The resulting course framework is the heart of this course and exam description and serves as a blueprint of the content and skills that can appear on an AP Exam.

The AP Test Development Committees are responsible for developing each AP Exam and ensuring the exam questions are aligned to the course framework. The AP Exam development process is a multiyear endeavor; all AP Exams undergo extensive review, revision, piloting, and analysis to ensure that questions are accurate, fair, and valid and that there is an appropriate spread of difficulty across the questions.

Committee members are selected to represent a variety of perspectives and institutions (public and private, small and large schools and colleges) and a range of gender, racial/ethnic, and regional groups. A list of each subject's current AP Test Development Committee members is available on **apcentral.collegeboard.org**.

Throughout AP course and exam development, College Board gathers feedback from various stakeholders in both secondary schools and higher education institutions. This feedback is carefully considered to ensure that AP courses and exams are able to provide students with a college-level learning experience and the opportunity to demonstrate their qualifications for advanced placement or college credit.

How AP Exams Are Scored

The exam scoring process, like the course and exam development process, relies on the expertise of both AP teachers and college faculty. While multiple-choice questions are scored by machine, the performance task is scored by thousands of college faculty and expert AP teachers. The performance tasks are scored either at the annual AP Reading or online. All AP Readers are thoroughly trained, and their work is monitored throughout the Reading for fairness and consistency. In each subject, a highly respected college faculty member serves as Chief Faculty Consultant and, with the help of AP Readers in leadership positions, maintains the accuracy of the scoring standards.

Scores on the performance task are weighted and combined with the results of the computer-scored multiple-choice questions, and this raw score is converted into a composite AP score on a 1–5 scale.

AP Exams are **not** norm-referenced or graded on a curve. Instead, they are criterion-referenced, which means that every student who meets the criteria for an AP score of 2, 3, 4, or 5 will receive that score, no matter how many students do so. The criteria for the number of points students must earn on the AP Exam to receive scores of 3, 4, or 5—the scores that research consistently validates for credit and placement purposes—include:

- The number of points successful college students earn when their professors administer AP Exam questions to them.
- The number of points researchers have found to be predictive that an AP student will succeed when placed into a subsequent, higher-level college course.
- Achievement-level descriptions formulated by college faculty who review each AP Exam question.

Using and Interpreting AP Scores

The extensive work done by college faculty and AP teachers in the development of the course and exam and throughout the scoring process ensures that AP Exam scores accurately represent students' achievement in the equivalent college course. Frequent and regular research studies establish the validity of AP scores as follows:

AP Score	Credit Recommendation	College Grade Equivalent
5	Extremely well qualified	A
4	Well qualified	A–, B+, B
3	Qualified	B–, C+, C
2	Possibly qualified	n/a
1	No recommendation	n/a

© 2023 College Board

While colleges and universities are responsible for setting their own credit and placement policies, most private colleges and universities award credit and/or advanced placement for AP scores of 3 or higher. Additionally, most states in the US have adopted statewide credit policies that assure college credit for scores of 3 or higher at public colleges and universities. To confirm a specific college's AP credit/placement policy, a search engine is available at **apstudent.org/creditpolicies**.

BECOMING AN AP READER

Each June, thousands of AP teachers and college faculty members from around the world gather for seven days in multiple locations or from home to evaluate and score the performance task section of the AP Exams. Ninety-eight percent of surveyed educators who took part in the AP Reading say it was a positive experience.

There are many reasons to consider becoming an AP Reader, including opportunities to:

- **Bring positive changes to the classroom:** Surveys show that the vast majority of returning AP Readers—both high school and college educators—make improvements to the way they teach or score because of their experience at the AP Reading.
- **Gain in-depth understanding of AP Exam and AP scoring standards:** AP Readers gain exposure to the quality and depth of the responses from the entire pool of AP Exam takers and thus are better able to assess their students' work in the classroom.
- **Receive compensation:** AP Readers are compensated for their work during the Reading. Expenses, lodging, and meals are covered for Readers who travel.
- **Score from home:** AP Readers have online distributed scoring opportunities for certain subjects, including AP Computer Science Principles. Check **collegeboard.org/apreading** for details.
- **Earn Continuing Education Units (CEUs):** AP Readers earn professional development hours and CEUs that can be applied to PD requirements by states, districts, and schools.

How to Apply

Visit **collegeboard.org/apreading** for eligibility requirements and to start the application process.

© 2023 College Board

AP Resources and Supports

By completing a simple activation process at the start of the school year, teachers and students receive access to a robust set of classroom resources.

AP Classroom

AP Classroom is a dedicated online platform designed to support teachers and students throughout their AP experience. The platform includes a variety of powerful resources and tools to provide yearlong support to teachers and enable students to receive meaningful feedback on their progress.

BIG IDEA GUIDES

Appearing in this publication and on AP Classroom, these planning guides outline all required course content and skills. Each big idea guide organizes content into topics and provides tips on taking the AP Exam.

TOPIC QUESTIONS

Formative AP questions provide feedback to students on the areas where they need to focus and are designed to meet students where they are in the material. Topic Questions are best used for spot-checking student understanding while teaching the topics identified in the course framework. They can be used in class or as homework based on teacher preference. The questions can reveal misunderstandings and help teachers target content and skills to emphasize in lessons and help students understand why an answer is correct or incorrect. Because the Topic Questions are formative, the results of these assessments cannot be used to evaluate teacher effectiveness or assign letter grades to students, and any such misuses are grounds for losing school authorization to offer AP courses.*

AP QUESTION BANK

This online library of real AP Exam questions provides teachers with secure questions to use in their classrooms. Teachers can find questions indexed by course topics and skills, create customized tests, and assign them online or on paper. These tests enable students to practice and get feedback on each question.

* To report misuses, please call 877-274-6474 (International: +1-212-632-1781).

© 2023 College Board

Digital Activation

In order to teach an AP class and make sure students are registered to take the AP Exam, teachers must first complete the digital activation process. Digital activation gives students and teachers access to the available resources and gathers students' exam registration information online, eliminating most of the answer sheet bubbling that has added to testing time and fatigue.

AP teachers and students begin by signing in to **My AP** and completing a simple activation process at the start of the school year, which provides access to all AP resources, including AP Classroom and AP Digital Portfolio.

To complete digital activation:

- Teachers and students sign in to, or create, their College Board accounts.
- Teachers confirm that they have added the course they teach to their AP Course Audit account and have had it approved by their school's administrator.
- Teachers or AP Coordinators, depending on who the school has decided is responsible, set up class sections so students can access AP resources and have exams ordered on their behalf.
- Students join class sections with a join code provided by their teacher or AP Coordinator.
- Students will be asked for additional registration information upon joining their first class section, which eliminates the need for extensive answer sheet bubbling on exam day.

While the digital activation process takes a short time for teachers, students, and AP Coordinators to complete, overall it helps save time and provides the following additional benefits:

- **Access to AP resources and supports:** Teachers have access to resources specifically designed to support instruction and provide feedback to students throughout the school year as soon as activation is complete.
- **Streamlined exam ordering:** AP Coordinators can create exam orders from the same online class rosters that enable students to access resources. The Coordinator reviews, updates, and submits this information as the school's exam order in the fall.
- **Student registration labels:** For each student included in an exam order, schools will receive a set of personalized AP ID registration labels, which replaces the AP student pack. The AP ID connects a student's exam materials with the registration information they provided during digital activation, eliminating the need for pre-administration sessions and reducing time spent bubbling on exam day.
- **Targeted Instructional Planning Reports:** AP teachers will get Instructional Planning Reports (IPRs) that include data on each of their class sections automatically rather than relying on special codes optionally bubbled in on exam day.
- **Integration with AP Digital Portfolio:** Class sections and student enrollment information are automatically sent to the AP Digital Portfolio. This automatic integration replaces any process teachers have used previously. The AP Digital Portfolio is where students will upload performance tasks and submit them as final.

© 2023 College Board

Instructional Model

Whether teachers choose to develop their own curriculum or adopt an AP endorsed provider's curriculum, integrating AP resources throughout the course can help students develop skills and conceptual understandings. The instructional model outlined below shows possible ways to incorporate AP resources into the classroom.

Plan

Teachers may consider the following approaches as they plan their instruction.

- Review the overview at the start of each **big idea guide** to identify essential questions, conceptual understandings, and skills for each big idea.
- Use the **Big Idea at a Glance** table to identify related topics that build toward a common understanding across big ideas, and then group those topics into units or modules, ensuring appropriate sequencing, scaffolding, and pacing for students.
- Identify useful strategies in the **Instructional Approaches** section to help teach the concepts and skills.

Teach

When teaching, supporting resources can be used to build students' conceptual understanding and mastery of skills.

- Use the topic pages in the **big idea guides** to identify the required content.
- Integrate the content with a skill, considering any appropriate scaffolding.
- Employ any of the instructional strategies identified during the planning stage.
- Use the available resources on the topic pages to bring a variety of assets into the classroom.

Assess

Teachers can measure student understanding of the content and skills covered in the big idea and provide actionable feedback to students.

- While teaching each big idea, use AP Classroom to assign students the **Topic Questions** as homework or as an in-class task.
- Provide question-level feedback to students through answer rationales; provide topic- and skill-level feedback.
- Create additional practice opportunities using the **AP Question Bank**, and assign them through AP Classroom.

© 2023 College Board

About the AP Computer Science Principles Course

AP Computer Science Principles introduces students to the breadth of the field of computer science. In this course, students will learn to design and evaluate solutions and to apply computer science to solve problems through the development of algorithms and programs. They will incorporate abstraction into programs and use data to discover new knowledge. Students will also explain how computing innovations and computing systems, including the Internet, work, explore their potential impacts, and contribute to a computing culture that is collaborative and ethical. It is important to note that the AP Computer Science Principles course does not have a designated programming language. Teachers have the flexibility to choose a programming language(s) that is most appropriate for their students to use in the classroom.

College Course Equivalent

AP Computer Science Principles is equivalent to an introductory, college-level breadth course in computer science.

Prerequisites

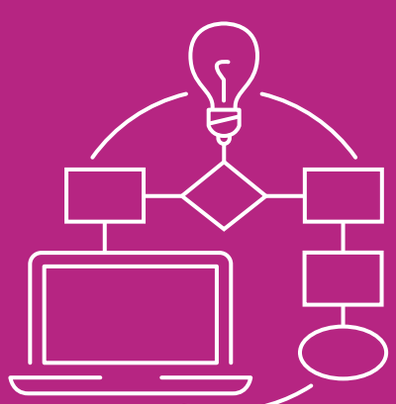
It is recommended that students in the AP Computer Science Principles course have successfully completed a first-year high school algebra course with a strong foundation of basic linear functions, composition of functions, and problem-solving strategies that require multiple approaches and collaborative efforts. In addition, students should be able to use a Cartesian (x, y) coordinate system to represent points on a plane. It is important that students and their advisers understand that any significant computer science course builds upon a foundation of mathematical reasoning that should be acquired before attempting such a course. Prior computer science experience is not required to take this course.

© 2023 College Board

THIS PAGE IS INTENTIONALLY LEFT BLANK.

AP COMPUTER SCIENCE PRINCIPLES

Course Framework



Introduction

Computer science involves problem-solving, hardware, and algorithms that help people utilize computers and incorporate multiple perspectives to address real-world problems in contemporary life. As the application of computer science is integrated into more aspects of our lives, it is important to understand the impact of computer science and how to maintain privacy, safety, and security not only when using computers but also while being the innovators of new computing applications. The course strives to engage all students, including those who have traditionally been underrepresented in computer science—such as female students, students of color, students with disabilities, and rural students—by allowing them to discover the power of computer science through rewarding yet challenging concepts.

A well-designed AP Computer Science Principles course that includes opportunities for students to collaborate to solve problems of their choice can help address traditional issues of equity and access. Such a course can broaden participation in computing while providing a strong and engaging introduction to the breadth of topics in the discipline.

The AP Computer Science Principles course reflects what computer science teachers, professors, and researchers have indicated are the main goals of an introductory, college-level computer science course:

- **Computational Solution Design**—Design and evaluate computational solutions for a purpose.
- **Algorithms and Program Development**—Develop and implement algorithms.
- **Abstraction in Program Development**—Develop programs that incorporate abstractions.
- **Code Analysis**—Evaluate and test algorithms and programs.
- **Computing Innovations**—Investigate computing innovations.
- **Responsible Computing**—Contribute to an inclusive, safe, collaborative, and ethical computing culture.

Students practice their computer science skills when designing and developing programs that address real-world problems and when investigating computing innovations they use or are interested in better understanding.

Compatible Curricula

The AP Computer Science Principles course surveys topics across several knowledge areas recommended by the Association for Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers Computing Society (IEEE-CS). Topics from the following knowledge areas have been included in this breadth-first approach to computer science: "Networking and Communication", "Parallel and Distributed Computing," "Software Development Fundamentals," "Programming Languages," "Architecture and Organization," "Computational Science," "Information Assurance and Security," and "Algorithms and Complexity." Teachers can review the **Computer Science Curricula from ACM and IEEE-CS** to see their complete curriculum guidelines.

© 2023 College Board

The AP Computer Science Principles course vertically aligns with the core concepts in the **Computer Science Teachers Association (CSTA) K–12 Computer Science Framework**. Most of the K–12 Core Practices and Concepts are extended in the AP Computer Science Principles course, and the expected depth of knowledge is significantly higher in the AP course. This vertical alignment allows CS teachers to make connections from earlier courses to the college-equivalent AP Computer Science Principles course.

AP Computer Science Program

AP Computer Science Principles is one of two AP Computer Science courses available to students. The AP Computer Science A course complements AP Computer Science Principles through a focused study of the Algorithms and Programming big idea. Students can take these two courses in either order or concurrently, as allowed by their school.

Resource Requirements

Students should have access to a computer system(s) that contains appropriate software to create and edit programs and allows students to practice, complete, and submit the AP Computer Science Principles performance task. The computer must have Internet access and be able to access the sites necessary for students to be successful in the course and assessment. The school ensures that each student has access to the AP Computer Science Principles Exam Reference Sheet (see Appendix), as well as performance task directions. The school ensures that each student has a college-level text or curricular resources deemed necessary by the teacher for individual use inside and outside of the classroom.

© 2023 College Board

Course Framework Components

Overview

This course framework provides a description of what students should know and be able to do to qualify for college credit or placement.

The course framework includes two essential components:

1 COMPUTATIONAL THINKING PRACTICES

The computational thinking practices are central to the study and practice of computer science. Students should practice and develop these skills on a regular basis over the span of the course.

2 COURSE CONTENT

The course content is organized into big ideas, which are cross-cutting concepts that build conceptual understanding and spiral throughout the course. The content and conceptual understandings within the big ideas reflect what colleges and universities typically expect students to master to qualify for college credit and/or placement.

© 2023 College Board

THIS PAGE IS INTENTIONALLY LEFT BLANK.

1

AP COMPUTER SCIENCE PRINCIPLES

Computational Thinking Practices

The table that follows presents the computational thinking practices that students should develop during the AP Computer Science Principles course. The practices form the basis of tasks on the AP Exam.

The learning objectives found in the big idea guides are each aligned to one of the skills from a practice. Teachers will want to be sure to integrate the practices and the course content with enough repetition to prepare students to transfer these skills when taking the AP Exam.

More detailed information about teaching the computational thinking practices can be found in the Instructional Approaches section of this publication.

© 2023 College Board

AP COMPUTER SCIENCE PRINCIPLES

Computational Thinking Practices: Skills

Practice 1	Practice 2	Practice 3	Practice 4	Practice 5	Practice 6
Computational Solution Design 1 Design and evaluate computational solutions for a purpose.	***Algorithms and Program Development*** 2 Develop and implement algorithms.	***Abstraction in Program Development*** 3 Develop programs that incorporate abstractions.	***Code Analysis*** 4 Evaluate and test algorithms and programs.	***Computing Innovations*** 5 Investigate computing innovations.	***Responsible Computing*** 6 Contribute to an inclusive, safe, collaborative, and ethical computing culture.
SKILLS					
1.A Investigate the situation, context, or task. 1.B Determine and design an appropriate method or approach to achieve the purpose. 1.C Explain how collaboration affects the development of a solution. 1.D Evaluate solution options.	2.A Represent algorithmic processes without using a programming language. 2.B Implement and apply an algorithm.	3.A Generalize data sources through variables. 3.B Use abstraction to manage complexity in a program. 3.C Explain how abstraction manages complexity.	4.A Explain how a code segment or program functions. 4.B Determine the result of code segments. 4.C Identify and correct errors in algorithms and programs, including error discovery through testing.	5.A Explain how computing systems work. 5.B Explain how knowledge can be generated from data. 5.C Describe the impact of a computing innovation. 5.D Describe the impact of gathering data. 5.E Evaluate the use of computing based on legal and ethical factors.	6.A Collaborate in the development of solutions. 6.B Use safe and secure methods when using computing devices. 6.C Acknowledge the intellectual property of others.

***NOTE:** All computational thinking practices except Computational Thinking Practice 6 are assessed in the multiple-choice section of the AP Exam.

© 2023 College Board

2

AP COMPUTER SCIENCE PRINCIPLES

Course Content

Based on the Understanding by Design® (Wiggins and McTighe) model, this course framework provides a description of the course requirements necessary for student success, with a focus on big ideas that encompass core principles, theories, and processes of the discipline. The framework also encourages instruction that prepares students for advanced computer science coursework and its integration into a wide array of STEM-related fields.

Big Ideas

The big ideas serve as the foundation of the course and help students create meaningful connections among concepts. They are often overarching concepts or themes that become threads that run throughout the course. Revisiting the big ideas and applying them in a variety of contexts enables students to develop deeper conceptual understanding. Below are the big ideas of the course and a brief description of each.

BIG IDEA 1: CREATIVE DEVELOPMENT (CRD)

When developing computing innovations, developers can use a formal, iterative design process or a less rigid process of experimentation. While using either approach, developers will encounter phases of investigating and reflecting, designing, prototyping, and testing. Additionally, collaboration is an important tool at any phase of development, because considering multiple perspectives allows for improvement of innovations.

BIG IDEA 2: DATA (DAT)

Data are central to computing innovations because they communicate initial conditions to programs and represent new knowledge. Computers consume data, transform data, and produce new data, allowing users to create new information or knowledge to solve problems through the interpretation of those data. Computers store data digitally, which means that the data must be manipulated in order to be presented in a useful way to the user.

continued on next page

© 2023 College Board

BIG IDEA 3: ALGORITHMS AND PROGRAMMING (AAP)

Programmers integrate algorithms and abstraction to create programs for creative purposes and to solve problems. Using multiple program statements in a specified order, making decisions, and repeating the same process multiple times are the building blocks of programs. Incorporating elements of abstraction—by breaking problems down into interacting pieces, each with their own purpose—makes writing complex programs easier. Programmers need to think algorithmically and use abstraction to define and interpret processes that are used in a program.

BIG IDEA 4: COMPUTING SYSTEMS AND NETWORKS (CSN)

Computer systems and networks are used to transfer data. One of the largest and most commonly used networks is the Internet. Through a series of protocols, the Internet can be used to send and receive information and ideas throughout the world. Transferring and processing information can be slow when done on a single computer, but leveraging multiple computers to do the work at the same time can significantly shorten the time it takes to complete tasks or solve problems.

BIG IDEA 5: IMPACT OF COMPUTING (IOC)

Computers and computing have revolutionized our lives. To use computing safely and responsibly, we need to be aware of privacy, security, and ethical issues. As programmers, we need to understand the potential impacts of our programs and be responsible for the consequences. As computer users, we need to understand any potential beneficial or harmful effects and how to protect ourselves and our privacy when using a computer.

© 2023 College Board

The five big ideas in AP Computer Science Principles, and their weighting on the multiple-choice section of the AP Exam, are listed below.

TOPICS

Each big idea is broken down into teachable segments called *topics*. The topic pages (starting on page 32) contain all required content for each topic. Although most topics can be taught in one or two class periods, teachers are encouraged to pace their course to suit the needs of their students and school.

Big Ideas	Exam Weighting
Big Idea 1: Creative Development	**10–13%**
Big Idea 2: Data	**17–22%**
Big Idea 3: Algorithms and Programming	**30–35%**
Big Idea 4: Computer Systems and Networks	**11–15%**
Big Idea 5: Impact of Computing	**21–26%**

© 2023 College Board

Course at a Glance

Plan

The Course at a Glance provides a useful visual organization of the AP Computer Science Principles curricular components, including the following:

- Big ideas, along with approximate weighting
- Progression of topics within each big idea
- Spiraling of practices across big ideas

This Course at a Glance is organized by big ideas rather than units of instruction. Within each big idea are topics. Teachers and AP endorsed providers can group topics together to create units or modules.

Teach

COMPUTATIONAL THINKING PRACTICES

Practices spiral across big ideas.

1. Computational Solution Design
2. Algorithms and Program Development
3. Abstraction in Program Development
4. Code Analysis
5. Computing Innovations
6. Responsible Computing

Assess

Assign the Topic Questions—either as homework or in class—for each big idea. The Topic Questions are formative AP questions that provide feedback to students on the areas where they need to focus.

BIG IDEA 1 Creative Development

10–13% AP Exam Weighting

- 1, 6 — **1.1** Collaboration
- 1, 3, 4 — **1.2** Program Function and Purpose
- 1, 4, 6 — **1.3** Program Design and Development
- 1, 4 — **1.4** Identifying and Correcting Errors

Topic Questions

Multiple-choice: ~20 questions

BIG IDEA 2 Data

17–22% AP Exam Weighting

- 1, 2, 3 — **2.1** Binary Numbers
- 1 — **2.2** Data Compression
- 5 — **2.3** Extracting Information from Data
- 2, 5 — **2.4** Using Programs with Data

Topic Questions

Multiple-choice: ~20 questions

© 2023 College Board

BIG IDEA 3 Algorithms and Programming

30–35% AP Exam Weighting

3 4	3.1 Variables and Assignments
3	3.2 Data Abstraction
2 4	3.3 Mathematical Expressions
4	3.4 Strings
2 4	3.5 Boolean Expressions
2 4	3.6 Conditionals
2 4	3.7 Nested Conditionals
2 4	3.8 Iteration
1 2	3.9 Developing Algorithms
2 4	3.10 Lists
1	3.11 Binary Search
3 4	3.12 Calling Procedures
3	3.13 Developing Procedures
2	3.14 Libraries
2 4	3.15 Random Values
1	3.16 Simulations
1	3.17 Algorithmic Efficiency
1	3.18 Undecidable Problems

Topic Questions

Multiple-choice: ~90 questions
Performance Task: ~20 prompts

BIG IDEA 4 Computer Systems and Networks

11-15% AP Exam Weighting

5	4.1 The Internet
1 5	4.2 Fault Tolerance
1	4.3 Parallel and Distributed Computing

Topic Questions

Multiple-choice: ~10 questions

BIG IDEA 5 Impact of Computing

21–26% AP Exam Weighting

5	5.1 Beneficial and Harmful Effects
5	5.2 Digital Divide
5	5.3 Computing Bias
1	5.4 Crowdsourcing
5	5.5 Legal and Ethical Concerns
5	5.6 Safe Computing

Topic Questions

Multiple-choice: ~20 questions

© 2023 College Board

THIS PAGE IS INTENTIONALLY LEFT BLANK.

AP COMPUTER SCIENCE PRINCIPLES

Big Idea Guides

Designed with input from the community of AP Computer Science Principles educators, the big idea guides offer all teachers helpful guidance in building students' skills and knowledge. It is important to remember that big ideas are themes that run across the entire school year, rather than units of instruction. Within each big idea are topics. Teachers and AP endorsed providers can group topics together to create units or modules for instruction. In some cases, it may be appropriate for a topic to appear in more than one unit or module.

A benefit of topics is that they enable the AP Program to provide interested teachers with formative Topic Questions that they can assign to their students, either throughout a unit or module or at the end, to gauge progress toward success on the AP Exam.

© 2023 College Board

Using the Big Idea Guides

BIG IDEA 1

10–13% AP EXAM WEIGHTING

Creative Development

ESSENTIAL QUESTIONS

CRD-1

- How has working collaboratively with other students improved an overall project?
- What are some ways you can collect additional feedback on your program to use for improvements?

CRD-2

- What are some ways you currently plan your work before starting a project?
- What apps or programs have you stopped using because you didn't like the design of how you interacted with it?

Developing Understanding

Collaboration is crucial when developing computing innovations, because having multiple perspectives offers opportunities to improve the design of innovations. In this big idea, students work collaboratively to design and develop programs using an iterative development process. They identify the needs of all users by gathering input from people from different backgrounds and demographics. Once the program is developed, they test it to ensure it meets these needs.

Effective collaboration can often differ from group work, because it requires equal participation and voice from all members of the group. Early in the school year, it may be helpful for teachers to establish practices and norms that facilitate a collaborative environment and provide students with time to practice working together. Content in this big idea is often paired with Big Idea 3: Algorithms and Programming.

Building Computational Thinking Practices

1.B 3.A 4.A 4.C

When designing a solution to a problem, programmers consider both the program itself and the way the user will interact with the program: the user interface. A well-designed user interface makes it easy for the user to understand what data are required as input for the program to complete its tasks.

When creating diagrams of their programs, students will benefit from considering how they want their program to behave based on identified inputs. Planning ahead may help them determine what abstractions can be developed and can help identify logic errors early in development.

When implementing program design, programmers often use documentation to explain the purpose of various code segments and describe how they function together in the program. Students' diagrams can be a good place to start writing their documentation. It may be more helpful to concentrate on documenting smaller code segments rather than trying to describe larger sections all at once.

Preparing for the AP Exam

Students will be expected to design and implement a program of their choice for the Create performance task. While students select their own topic for this task, they are required to include certain elements, such as lists and procedures, in their program code. Providing students with exemplars may help them consider the types of programs that can be developed while still meeting this requirement.

Students will need practice identifying and correcting errors to prepare for the AP Exam. One way to give students this practice is to provide them with prewritten program code to correct.

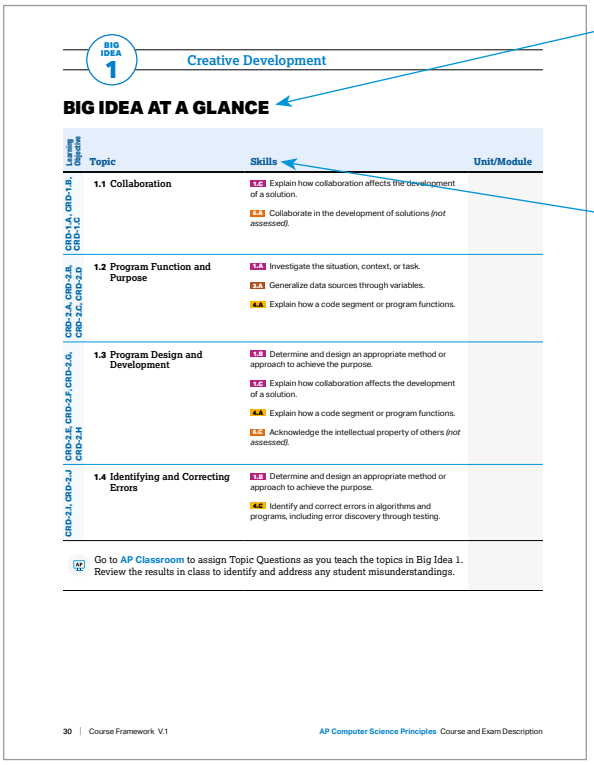
BIG IDEA OPENERS

The **Essential Questions** are thought-provoking questions that motivate students and inspire inquiry. They are organized by the enduring understandings that appear in the big idea.

Developing Understanding provides an overview that contextualizes and situates the key content of the big idea within the scope of the course. **Big Ideas** are overarching concepts or themes that become threads that run throughout the course.

Building Computational Thinking Practices describes specific skills within the practices that are appropriate to focus on in that big idea.

Preparing for the AP Exam provides helpful tips and common student misunderstandings identified from prior exam data.



BIG IDEA 1 Creative Development

BIG IDEA AT A GLANCE

Learning Objective	Topic	Skills	Unit/Module
CRD-1.A, CRD-1.B, CRD-1.C	1.1 Collaboration	1.C Explain how collaboration affects the development of a solution. 2.J Collaborate in the development of solutions *(not assessed)*.	
CRD-2.A, CRD-2.B, CRD-2.E, CRD-2.D	1.2 Program Function and Purpose	1.A Investigate the situation, context, or task. 3.A Generalize data sources through variables. 4.A Explain how a code segment or program functions.	
CRD-2.E, CRD-2.F, CRD-2.G, CRD-2.H	1.3 Program Design and Development	1.B Determine and design an appropriate method or approach to achieve the purpose. 1.C Explain how collaboration affects the development of a solution. 4.A Explain how a code segment or program functions. 6.C Acknowledge the intellectual property of others *(not assessed)*.	
CRD-2.I, CRD-2.J	1.4 Identifying and Correcting Errors	1.B Determine and design an appropriate method or approach to achieve the purpose. 4.C Identify and correct errors in algorithms and programs, including error discovery through testing.	

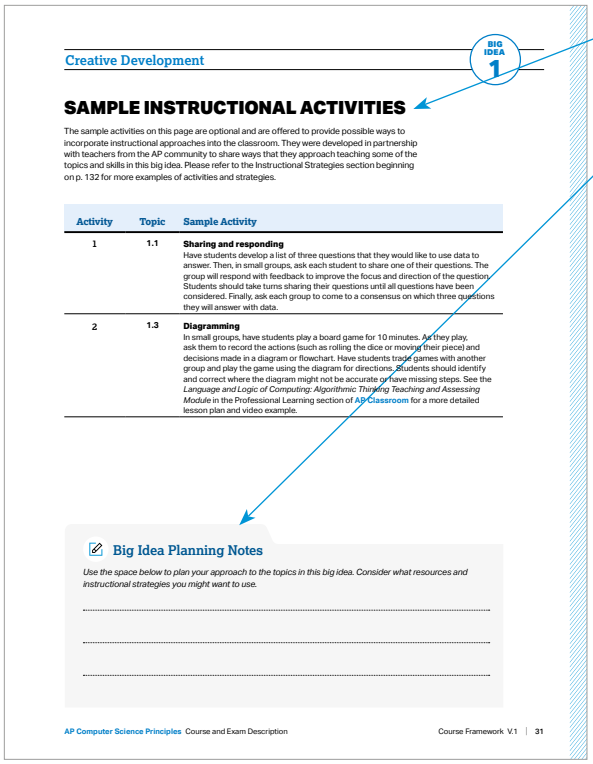
Go to AP Classroom to assign Topic Questions as you teach the topics in Big Idea 1. Review the results in class to identify and address any student misunderstandings.

The **Big Idea at a Glance** table shows the topics and related learning objectives and skills. The Unit/Module column has been left blank so that teachers can indicate where they are going to place each topic in their course. Teachers who are using an AP endorsed provider can determine where each topic is being taught in their provider's curriculum.

The **Skills** for each topic show how the content in that topic has been linked to specific AP Computer Science Principles skills.

© 2023 College Board

Using the Big Idea Guides



Creative Development

BIG IDEA 1

SAMPLE INSTRUCTIONAL ACTIVITIES

The sample activities on this page are optional and are offered to provide possible ways to incorporate instructional approaches into the classroom. They were developed in partnership with teachers from the AP community to share ways that they approach teaching some of the topics and skills in this big idea. Please refer to the Instructional Strategies section beginning on p. 132 for more examples of activities and strategies.

Activity	Topic	Sample Activity
1	1.1	**Sharing and responding** Have students develop a list of three questions that they would like to use data to answer. Then, in small groups, ask each student to share one of their questions. The group will respond with feedback to improve the focus and direction of the question. Students should take turns sharing their questions until all questions have been considered. Finally, ask each group to come to a consensus on which three questions they will answer with data.
2	1.3	**Diagramming** In small groups, have students play a board game for 10 minutes. As they play, ask them to record the actions (such as rolling the dice or moving their piece) and decisions made in a diagram or flowchart. Have students trade games with another group and play the game using the diagram for directions. Students should identify and correct where the diagram might not be accurate or have missing steps. See the *Language and Logic of Computing: Algorithmic Thinking Teaching and Assessing Module* in the Professional Learning section of AP Classroom for a more detailed lesson plan and video example.

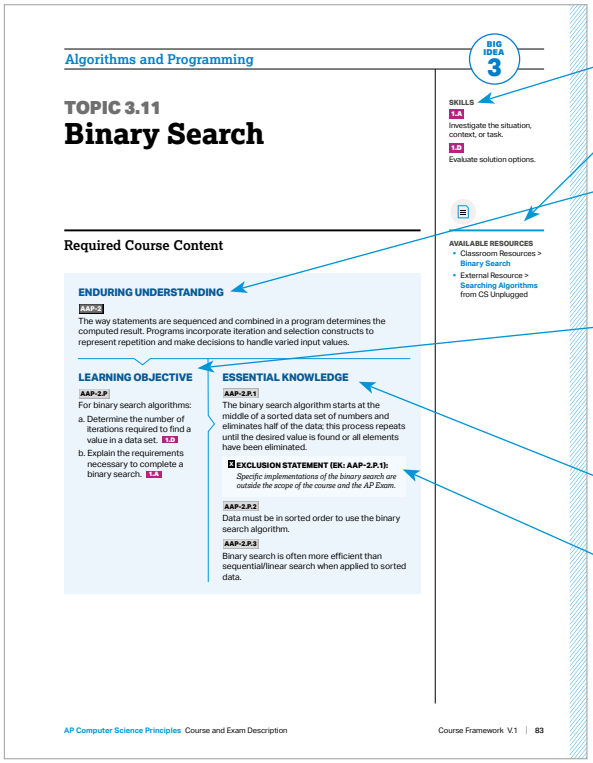
Big Idea Planning Notes

Use the space below to plan your approach to the topics in this big idea. Consider what resources and instructional strategies you might want to use.

AP Computer Science Principles Course and Exam Description — Course Framework V.1 | 31

The **Sample Instructional Activities** page includes optional activities that can help teachers tie together the content and skill of a particular topic.

The **Big Idea Planning Notes** section provides space for teachers to make notes on their approach to the individual topics and the big idea as a whole.



Algorithms and Programming

BIG IDEA 3

TOPIC 3.11

Binary Search

SKILLS
1.A Investigate the situation, context, or task.
1.D Evaluate solution options.

AVAILABLE RESOURCES
- Classroom Resources > Binary Search
- External Resource > Searching Algorithms from CS Unplugged

Required Course Content

ENDURING UNDERSTANDING
AAP-2
The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.

LEARNING OBJECTIVE
AAP-2.P
For binary search algorithms:
a. Determine the number of iterations required to find a value in a data set. 1.D
b. Explain the requirements necessary to complete a binary search. 1.A

ESSENTIAL KNOWLEDGE
AAP-2.P.1
The binary search algorithm starts at the middle of a sorted data set of numbers and eliminates half of the data; this process repeats until the desired value is found or all elements have been eliminated.

EXCLUSION STATEMENT (EK: AAP-2.P.1):
Specific implementations of the binary search are outside the scope of the course and the AP Exam.

AAP-2.P.2
Data must be in sorted order to use the binary search algorithm.

AAP-2.P.3
Binary search is often more efficient than sequential/linear search when applied to sorted data.

AP Computer Science Principles Course and Exam Description — Course Framework V.1 | 83

TOPIC PAGES

The **Skills** section offers one or more skills related to the topic.

Where possible, **available resources** are listed that might help teachers address a particular topic in their classroom.

Enduring understandings are the long-term takeaways related to the big ideas that leave a lasting impression on students. Students build and earn these understandings over time by exploring and applying course content throughout the year.

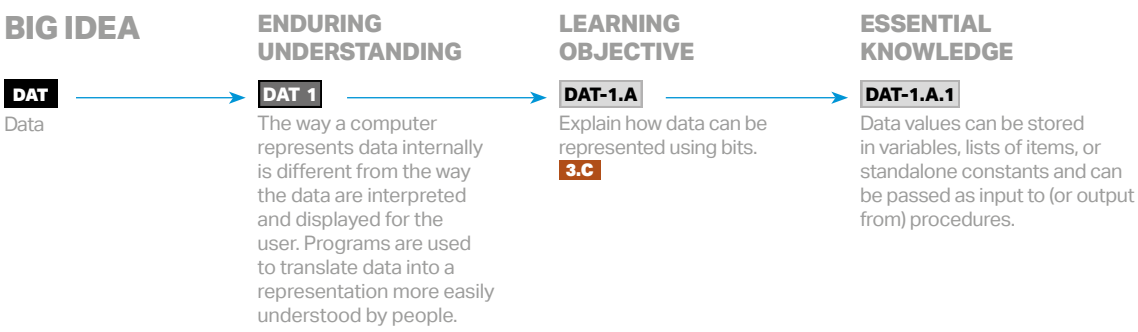
Learning objectives define what a student should be able to do with content knowledge to progress toward the enduring understandings. Each learning objective is paired with a recommended skill, and that skill appears in a colored box after the learning objective.

Essential knowledge statements describe the knowledge required to perform the learning objective.

Exclusion statements define content or specific details about content that does not need to be included in the course. The content in the exclusion statements will not be assessed on the AP Computer Science Principles Exam.

© 2023 College Board

REQUIRED COURSE CONTENT LABELING SYSTEM



NOTE: Labels are used to distinguish each element of the required course content and are used throughout this course and exam description. Additionally, they are used in the AP Question Bank and other resources found on AP Classroom. Enduring understandings are labeled sequentially according to the big idea that they are related to. Learning objectives are labeled to correspond with the enduring understanding they relate to. Finally, essential knowledge statements are labeled to correspond with the learning objective they relate to.

© 2023 College Board

AP COMPUTER SCIENCE PRINCIPLES

BIG IDEA 1

Creative Development

10–13%
AP EXAM WEIGHTING

© 2023 College Board

Remember to go to **AP Classroom** to assign students the online Topic Questions for this big idea.

Whether assigned as homework or completed in class, the Topic Questions can be used to spot-check student understanding and help identify content and skills to emphasize in lessons.

Topic Questions

Multiple-choice: ~20 questions

© 2023 College Board

BIG IDEA 1

10–13% AP EXAM WEIGHTING

Creative Development

ESSENTIAL QUESTIONS

CRD-1

- How has working collaboratively with other students improved an overall project?
- What are some ways you can collect additional feedback on your program to use for improvements?

CRD 2

- What are some ways you currently plan your work before starting a project?
- What apps or programs have you stopped using because you didn't like the design of how you interacted with it?

Developing Understanding

Collaboration is crucial when developing computing innovations, because having multiple perspectives offers opportunities to improve the design of innovations. In this big idea, students work collaboratively to design and develop programs using an iterative development process. They identify the needs of all users by gathering input from people from different backgrounds and demographics. Once the program is developed, they test it to ensure it meets these needs.

Effective collaboration can often differ from group work, because it requires equal participation and voice from all members of the group. Early in the school year, it may be helpful for teachers to establish practices and norms that facilitate a collaborative environment and provide students with time to practice working together. Content in this big idea is often paired with Big Idea 3: Algorithms and Programming.

Building Computational Thinking Practices

1.B 3.A 4.A 4.C

When designing a solution to a problem, programmers consider both the program itself and the way the user will interact with the program: the user interface. A well-designed user interface makes it easy for the user to understand what data are required as input for the program to complete its tasks.

When creating diagrams of their programs, students will benefit from considering how they want their program to behave based on identified inputs. Planning ahead may help them determine what abstractions can be developed and can help identify logic errors early in development.

When implementing program design, programmers often use documentation to explain the purpose of various code segments and describe how they function together in the program. Students' diagrams can be a good place to start writing their documentation. It may be more helpful to concentrate on documenting smaller code segments rather than trying to describe larger sections all at once.

Preparing for the AP Exam

Students will be expected to design and implement a program of their choice for the Create performance task. While students select their own topic for this task, they are required to include certain elements, such as lists and procedures, in their program code. Providing students with exemplars may help them consider the types of programs that can be developed while still meeting this requirement.

Students will need practice identifying and correcting errors to prepare for the AP Exam. One way to give students this practice is to provide them with prewritten program code to correct.

© 2023 College Board

BIG IDEA AT A GLANCE

Learning Objective	Topic	Skills	Unit/Module
CRD-1.A, CRD-1.B, CRD-1.C	**1.1 Collaboration**	1.C Explain how collaboration affects the development of a solution. 6.A Collaborate in the development of solutions *(not assessed)*.	
CRD-2.A, CRD-2.B, CRD-2.C, CRD-2.D	**1.2 Program Function and Purpose**	1.A Investigate the situation, context, or task. 3.A Generalize data sources through variables. 4.A Explain how a code segment or program functions.	
CRD-2.E, CRD-2.F, CRD-2.G, CRD-2.H	**1.3 Program Design and Development**	1.B Determine and design an appropriate method or approach to achieve the purpose. 1.C Explain how collaboration affects the development of a solution. 4.A Explain how a code segment or program functions. 6.C Acknowledge the intellectual property of others *(not assessed)*.	
CRD-2.I, CRD-2.J	**1.4 Identifying and Correcting Errors**	1.B Determine and design an appropriate method or approach to achieve the purpose. 4.C Identify and correct errors in algorithms and programs, including error discovery through testing.	

Go to **AP Classroom** to assign Topic Questions as you teach the topics in Big Idea 1. Review the results in class to identify and address any student misunderstandings.

© 2023 College Board

SAMPLE INSTRUCTIONAL ACTIVITIES

The sample activities on this page are optional and are offered to provide possible ways to incorporate instructional approaches into the classroom. They were developed in partnership with teachers from the AP community to share ways that they approach teaching some of the topics and skills in this big idea. Please refer to the Instructional Strategies section beginning on p. 132 for more examples of activities and strategies.

Activity	Topic	Sample Activity
1	1.1	**Sharing and responding** Have students develop a list of three questions that they would like to use data to answer. Then, in small groups, ask each student to share one of their questions. The group will respond with feedback to improve the focus and direction of the question. Students should take turns sharing their questions until all questions have been considered. Finally, ask each group to come to a consensus on which three questions they will answer with data.
2	1.3	**Diagramming** In small groups, have students play a board game for 10 minutes. As they play, ask them to record the actions (such as rolling the dice or moving their piece) and decisions made in a diagram or flowchart. Have students trade games with another group and play the game using the diagram for directions. Students should identify and correct where the diagram might not be accurate or have missing steps. See the *Language and Logic of Computing: Algorithmic Thinking Teaching and Assessing Module* in the Professional Learning section of **AP Classroom** for a more detailed lesson plan and video example.

Big Idea Planning Notes

Use the space below to plan your approach to the topics in this big idea. Consider what resources and instructional strategies you might want to use.

© 2023 College Board

SKILLS

1.C
Explain how collaboration affects the development of a solution.

6.A
Collaborate in the development of solutions.

AVAILABLE RESOURCES

- External Resources >
  - **Collaboration Tools** from Cornell University Center for Teaching Innovation
  - **4 Methods to Enhance Student Collaboration in the Classroom** from Concordia University-Portland

TOPIC 1.1
Collaboration

Required Course Content

ENDURING UNDERSTANDING

CRD-1
Incorporating multiple perspectives through collaboration improves computing innovations as they are developed.

LEARNING OBJECTIVE

CRD-1.A
Explain how computing innovations are improved through collaboration. 1.C

ESSENTIAL KNOWLEDGE

CRD-1.A.1
A computing innovation includes a program as an integral part of its function.

CRD-1.A.2
A computing innovation can be physical (e.g., self-driving car), nonphysical computing software (e.g., picture editing software), or a nonphysical computing concept (e.g., e-commerce).

CRD-1.A.3
Effective collaboration produces a computing innovation that reflects the diversity of talents and perspectives of those who designed it.

CRD-1.A.4
Collaboration that includes diverse perspectives helps avoid bias in the development of computing innovations.

CRD-1.A.5
Consultation and communication with users are important aspects of the development of computing innovations.

continued on next page

© 2023 College Board

Creative Development

LEARNING OBJECTIVE	ESSENTIAL KNOWLEDGE
CRD-1.A Explain how computing innovations are improved through collaboration. 1.C	**CRD-1.A.6** Information gathered from potential users can be used to understand the purpose of a program from diverse perspectives and to develop a program that fully incorporates these perspectives.
CRD-1.B Explain how computing innovations are developed by groups of people. 1.C	**CRD-1.B.1** Online tools support collaboration by allowing programmers to share and provide feedback on ideas and documents. **CRD-1.B.2** Common models such as pair programming exist to facilitate collaboration.
CRD-1.C Demonstrate effective interpersonal skills during collaboration. 1.C	**CRD-1.C.1** Effective collaborative teams practice interpersonal skills, including but not limited to: ▪ communication ▪ consensus building ▪ conflict resolution ▪ negotiation

© 2023 College Board

SKILLS

1.A
Investigate the situation, context, or task.

3.A
Generalize data sources through variables.

4.A
Explain how a code segment or program functions.

AVAILABLE RESOURCES

- Professional Development > **Teaching and Assessing Module: Explaining Processes**
- External Resource > **ACM Tech News**

TOPIC 1.2
Program Function and Purpose

Required Course Content

ENDURING UNDERSTANDING

CRD-2
Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking.

LEARNING OBJECTIVE	ESSENTIAL KNOWLEDGE
CRD-2.A Describe the purpose of a computing innovation. **1.A**	**CRD-2.A.1** The purpose of computing innovations is to solve problems or to pursue interests through creative expression.
	CRD-2.A.2 An understanding of the purpose of a computing innovation provides developers with an improved ability to develop that computing innovation.
CRD-2.B Explain how a program or code segment functions. **4.A**	**CRD-2.B.1** A *program* is a collection of program statements that performs a specific task when run by a computer. A program is often referred to as *software*.
	CRD-2.B.2 A *code segment* is a collection of program statements that is part of a program.
	CRD-2.B.3 A program needs to work for a variety of inputs and situations.
	CRD-2.B.4 The *behavior* of a program is how a program functions during execution and is often described by how a user interacts with it.

continued on next page

© 2023 College Board

LEARNING OBJECTIVE	ESSENTIAL KNOWLEDGE
CRD-2.B Explain how a program or code segment functions. 4.A	**CRD-2.B.5** A program can be described broadly by what it does, or in more detail by both what the program does and how the program statements accomplish this function.
CRD-2.C Identify input(s) to a program. 3.A	**CRD-2.C.1** *Program inputs* are data sent to a computer for processing by a program. Input can come in a variety of forms, such as tactile, audio, visual, or text. **CRD-2.C.2** An *event* is associated with an action and supplies input data to a program. **CRD-2.C.3** Events can be generated when a key is pressed, a mouse is clicked, a program is started, or any other defined action occurs that affects the flow of execution. **CRD-2.C.4** Inputs usually affect the output produced by a program. **CRD-2.C.5** In event-driven programming, program statements are executed when triggered rather than through the sequential flow of control. **CRD-2.C.6** Input can come from a user or other programs.
CRD-2.D Identify output(s) produced by a program. 3.A	**CRD-2.D.1** *Program outputs* are any data sent from a program to a device. Program output can come in a variety of forms, such as tactile, audio, visual, or text. **CRD-2.D.2** Program output is usually based on a program's input or prior state (e.g., internal values).

© 2023 College Board

SKILLS

1.B
Determine and design an appropriate method or approach to achieve the purpose.

1.C
Explain how collaboration affects the development of a solution.

4.A
Explain how a code segment or program functions.

6.C
Acknowledge the intellectual property of others.

AVAILABLE RESOURCE

- Professional Development > Teaching and Assessing Module: Explaining Processes

TOPIC 1.3
Program Design and Development

Required Course Content

ENDURING UNDERSTANDING

CRD-2

Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking.

LEARNING OBJECTIVE

CRD-2.E

Develop a program using a development process. 1.B

ESSENTIAL KNOWLEDGE

CRD-2.E.1

A development process can be ordered and intentional, or exploratory in nature.

CRD-2.E.2

There are multiple development processes. The following phases are commonly used when developing a program:

- investigating and reflecting
- designing
- prototyping
- testing

CRD-2.E.3

A development process that is iterative requires refinement and revision based on feedback, testing, or reflection throughout the process. This may require revisiting earlier phases of the process.

CRD-2.E.4

A development process that is incremental is one that breaks the problem into smaller pieces and makes sure each piece works before adding it to the whole.

continued on next page

© 2023 College Board

Creative Development

LEARNING OBJECTIVE

CRD-2.F

Design a program and its user interface. **1.B**

ESSENTIAL KNOWLEDGE

CRD-2.F.1

The design of a program incorporates investigation to determine its requirements.

CRD-2.F.2

Investigation in a development process is useful for understanding and identifying the program constraints, as well as the concerns and interests of the people who will use the program.

CRD-2.F.3

Some ways investigation can be performed are as follows:

- collecting data through surveys
- user testing
- interviews
- direct observations

CRD-2.F.4

Program requirements describe how a program functions and may include a description of user interactions that a program must provide.

CRD-2.F.5

A program's specification defines the requirements for the program.

CRD-2.F.6

In a development process, the design phase outlines how to accomplish a given program specification.

CRD-2.F.7

The design phase of a program may include:

- brainstorming
- planning and storyboarding
- organizing the program into modules and functional components
- creation of diagrams that represent the layouts of the user interface
- development of a testing strategy for the program

continued on next page

© 2023 College Board

Creative Development

LEARNING OBJECTIVE	ESSENTIAL KNOWLEDGE
CRD-2.G Describe the purpose of a code segment or program by writing documentation. **4.A**	**CRD-2.G.1** *Program documentation* is a written description of the function of a code segment, event, procedure, or program and how it was developed.
	CRD-2.G.2 *Comments* are a form of program documentation written into the program to be read by people and do not affect how a program runs.
	CRD-2.G.3 Programmers should document a program throughout its development.
	CRD-2.G.4 Program documentation helps in developing and maintaining correct programs when working individually or in collaborative programming environments.
	CRD-2.G.5 Not all programming environments support comments, so other methods of documentation may be required.
CRD-2.H Acknowledge code segments used from other sources. **1.C**	**CRD-2.H.1** It is important to acknowledge any code segments that were developed collaboratively or by another source.
	CRD-2.H.2 Acknowledgement of a code segment(s) written by someone else and used in a program can be in the program documentation. The acknowledgement should include the origin or original author's name.

© 2023 College Board

TOPIC 1.4
Identifying and Correcting Errors

Required Course Content

ENDURING UNDERSTANDING

CRD-2

Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking.

LEARNING OBJECTIVE

CRD-2.I

For errors in an algorithm or program:

a. Identify the error. **4.C**

b. Correct the error. **4.C**

ESSENTIAL KNOWLEDGE

CRD-2.I.1

A *logic error* is a mistake in the algorithm or program that causes it to behave incorrectly or unexpectedly.

CRD-2.I.2

A *syntax error* is a mistake in the program where the rules of the programming language are not followed.

CRD-2.I.3

A *run-time error* is a mistake in the program that occurs during the execution of a program. Programming languages define their own run-time errors.

CRD-2.I.4

An *overflow error* is an error that occurs when a computer attempts to handle a number that is outside of the defined range of values.

CRD-2.I.5

The following are effective ways to find and correct errors:

- test cases
- hand tracing
- visualizations
- debuggers
- adding extra output statement(s)

continued on next page

SKILLS

1.B

Determine and design an appropriate method or approach to achieve the purpose.

4.C

Identify and correct errors in algorithms and programs, including error discovery through testing.

AVAILABLE RESOURCE

- Professional Development > **Teaching and Assessing Module: Explaining Processes**

© 2023 College Board

Creative Development

LEARNING OBJECTIVE

CRD-2.J

Identify inputs and corresponding expected outputs or behaviors that can be used to check the correctness of an algorithm or program. **4.C**

ESSENTIAL KNOWLEDGE

CRD-2.J.1

In the development process, *testing* uses defined inputs to ensure that an algorithm or program is producing the expected outcomes. Programmers use the results from testing to revise their algorithms or programs.

CRD-2.J.2

Defined inputs used to test a program should demonstrate the different expected outcomes that are at or just beyond the extremes (minimum and maximum) of input data.

CRD-2.J.3

Program requirements are needed to identify appropriate defined inputs for testing.

© 2023 College Board

AP COMPUTER SCIENCE PRINCIPLES

BIG IDEA 2

Data

17–22%
AP EXAM WEIGHTING

© 2023 College Board

Remember to go to **AP Classroom** to assign students the online Topic Questions for this big idea.

Whether assigned as homework or completed in class, the Topic Questions can be used to spot-check student understanding and help identify content and skills to emphasize in lessons.

Topic Questions

Multiple-choice: ~20 questions

© 2023 College Board

BIG IDEA 2

17–22% AP EXAM WEIGHTING

Data

ESSENTIAL QUESTIONS

DAT-1

- How can we use 1s and 0s to represent something complex like a video of the marching band playing a song?

DAT-2

- How can you predict the attendance at a school event using data gathered from social media?
- When is it more appropriate to use a computer to analyze data than to complete the analysis by hand?

Developing Understanding

Because essentially everything we do with a computer is being broken down into some form of data, it is important for students to develop a breadth of understanding of how computers handle data and how students can use those same data to solve problems such as raising awareness for a cause, using census data to determine which state will gain seats in the House of Representatives, or using traffic and cost data to determine the ideal location for prom. In this big idea, students will gain a deep understanding of how information is stored on a computer in binary and seamlessly translated into what is seen on the screen or heard through speakers. Students will also learn how data are processed to learn something new. This big idea is often paired with Big Idea 3: Algorithms and Programming and Big Idea 5: Impact of Computing.

Building Computational Thinking Practices

2.B 3.C 5.B

On the end-of-course exam, students will be presented with the way data for text or media, such as color, are represented by a computer and will be asked to convert values from binary to decimal or vice versa. The idea that there are number systems other than the decimal system is often new to students. Connecting the foundational principles of how number systems operate to the decimal number system is likely to help students lean on their prior knowledge when asked to work with binary numbers.

Some real-world problems and simulations involve the use of large data sets that cannot be easily analyzed by hand and require a programming solution that manipulates or combines the data with other sources to generate new knowledge and find a solution. When working with large data sets, programmers use data abstraction to write programs that can be flexible enough to handle a change in the number of data entries. Providing students with practice using data sets that are too large to manipulate by hand will motivate them to develop more general solutions and data abstractions. Because the solution is generalized, an explanation of the solution through documentation within the program may be necessary.

Preparing for the AP Exam

Data compression algorithms are often used to maximize storage space or to transmit data over the Internet, but sometimes at a cost to the quality of the data. Students will need to compare data compression algorithms and determine which one would be best to use in a given situation. Providing students with examples of different compression algorithms along with how each might work may deepen students' understanding. Examples of how data might be restored to their uncompressed state—or may be unable to be restored—may improve students' ability to distinguish between compression algorithms.

When presented with scenarios that describe data and metadata for analysis on the end-of-course exam, students will be asked to determine what information can be found, as well as a potential programming process that can be used to extract information or modify the existing data. Students might benefit from practice identifying a problem they could solve using data, such as the best route to take to school, gathering the necessary data to analyze—either by using a public data set or developing a survey to gather the data—and then implementing a program that will manipulate the data to find an answer.

© 2023 College Board

BIG IDEA AT A GLANCE

Learning Objective	Topic	Skills	Unit/Module
DAT-1.A, DAT-1.B, DAT-1.C	**2.1 Binary Numbers**	1.D Evaluate solution options. 2.B Implement and apply an algorithm. 3.C Explain how abstraction manages complexity.	
DAT-1.D	**2.2 Data Compression**	1.D Evaluate solution options.	
DAT-2.A, DAT-2.B, DAT-2.C	**2.3 Extracting Information from Data**	5.B Explain how knowledge can be generated from data. 5.D Describe the impact of gathering data.	
DAT-2.D, DAT-2.E	**2.4 Using Programs with Data**	2.B Implement and apply an algorithm. 5.B Explain how knowledge can be generated from data.	

Go to **AP Classroom** to assign Topic Questions as you teach the topics in Big Idea 2. Review the results in class to identify and address any student misunderstandings.

© 2023 College Board

SAMPLE INSTRUCTIONAL ACTIVITIES

The sample activities on this page are optional and are offered to provide possible ways to incorporate instructional approaches into the classroom. They were developed in partnership with teachers from the AP community to share ways that they approach teaching some of the topics and skills in this big idea. Please refer to the Instructional Strategies section beginning on p. 132 for more examples of activities and strategies.

Activity	Topic	Sample Activity
1	2.1	**Look for a pattern** Provide students with a sentence or paragraph of compressed lossless text and a key. Have them look for patterns in their process of retrieving the original text and evaluate whether this is the best compression algorithm to use. Have them write down the patterns they see along with their evaluation and share these in a large group.
2	2.4	**Diagramming** Give students a question and a list of data. Have them diagram a process that could be used to answer the question using the data, making sure to include the input(s) of information and the output of the transformed data. Have students include an explanation of how the process represented in their diagram would work to find the solution.

Big Idea Planning Notes

Use the space below to plan your approach to the topics in this big idea. Consider what resources and instructional strategies you might want to use.

© 2023 College Board

SKILLS

1.D
Evaluate solution options.

2.B
Implement and apply an algorithm.

3.C
Explain how abstraction manages complexity.

AVAILABLE RESOURCES

- External Resources >
  - **Binary Numbers** from CS Unplugged
  - **Blown to Bits: Chapter 1**

TOPIC 2.1
Binary Numbers

Required Course Content

ENDURING UNDERSTANDING

DAT-1

The way a computer represents data internally is different from the way the data are interpreted and displayed for the user. Programs are used to translate data into a representation more easily understood by people.

LEARNING OBJECTIVE

DAT-1.A

Explain how data can be represented using bits. 3.C

ESSENTIAL KNOWLEDGE

DAT-1.A.1

Data values can be stored in variables, lists of items, or standalone constants and can be passed as input to (or output from) procedures.

DAT-1.A.2

Computing devices represent data digitally, meaning that the lowest-level components of any value are bits.

DAT-1.A.3

Bit is shorthand for *binary digit* and is either 0 or 1.

DAT-1.A.4

A *byte* is 8 bits.

DAT-1.A.5

Abstraction is the process of reducing complexity by focusing on the main idea. By hiding details irrelevant to the question at hand and bringing together related and useful details, abstraction reduces complexity and allows one to focus on the idea.

DAT-1.A.6

Bits are grouped to represent abstractions. These abstractions include, but are not limited to, numbers, characters, and color.

continued on next page

© 2023 College Board

LEARNING OBJECTIVE	ESSENTIAL KNOWLEDGE
DAT-1.A Explain how data can be represented using bits. 3.C	**DAT-1.A.7** The same sequence of bits may represent different types of data in different contexts. **DAT-1.A.8** Analog data have values that change smoothly, rather than in discrete intervals, over time. Some examples of analog data include pitch and volume of music, colors of a painting, or position of a sprinter during a race. **DAT-1.A.9** The use of digital data to approximate real-world analog data is an example of abstraction. **DAT-1.A.10** Analog data can be closely approximated digitally using a *sampling technique*, which means measuring values of the analog signal at regular intervals called *samples*. The samples are measured to figure out the exact bits required to store each sample.
DAT-1.B Explain the consequences of using bits to represent data. 1.D	**DAT-1.B.1** In many programming languages, integers are represented by a fixed number of bits, which limits the range of integer values and mathematical operations on those values. This limitation can result in overflow or other errors. **DAT-1.B.2** Other programming languages provide an abstraction through which the size of representable integers is limited only by the size of the computer's memory; this is the case for the language defined in the exam reference sheet. **DAT-1.B.3** In programming languages, the fixed number of bits used to represent real numbers limits the range and mathematical operations on these values; this limitation can result in round-off and other errors. Some real numbers are represented as approximations in computer storage. **EXCLUSION STATEMENT (EK DAT-1.B.3):** *Specific range limitations for real numbers are outside the scope of this course and the AP Exam.*

continued on next page

© 2023 College Board

LEARNING OBJECTIVE

DAT-1.C

For binary numbers:

a. Calculate the binary (base 2) equivalent of a positive integer (base 10) and vice versa. 2.B

b. Compare and order binary numbers. 2.B

ESSENTIAL KNOWLEDGE

DAT-1.C.1

Number bases, including binary and decimal, are used to represent data.

DAT-1.C.2

Binary (base 2) uses only combinations of the digits zero and one.

DAT-1.C.3

Decimal (base 10) uses only combinations of the digits 0 – 9.

DAT-1.C.4

As with decimal, a digit's position in the binary sequence determines its numeric value. The numeric value is equal to the bit's value (0 or 1) multiplied by the place value of its position.

DAT-1.C.5

The place value of each position is determined by the base raised to the power of the position. Positions are numbered starting at the rightmost position with 0 and increasing by 1 for each subsequent position to the left.

© 2023 College Board

BIG IDEA 2

TOPIC 2.2
Data Compression

SKILL

1.D

Evaluate solution options.

AVAILABLE RESOURCE

- External Resource > **Text Compression** from CS Unplugged

Required Course Content

ENDURING UNDERSTANDING

DAT-1

The way a computer represents data internally is different from the way the data are interpreted and displayed for the user. Programs are used to translate data into a representation more easily understood by people.

LEARNING OBJECTIVE

DAT-1.D

Compare data compression algorithms to determine which is best in a particular context. 1.D

ESSENTIAL KNOWLEDGE

DAT-1.D.1

Data compression can reduce the size (number of bits) of transmitted or stored data.

DAT-1.D.2

Fewer bits does not necessarily mean less information.

DAT-1.D.3

The amount of size reduction from compression depends on both the amount of redundancy in the original data representation and the compression algorithm applied.

DAT-1.D.4

Lossless data compression algorithms can usually reduce the number of bits stored or transmitted while guaranteeing complete reconstruction of the original data.

DAT-1.D.5

Lossy data compression algorithms can significantly reduce the number of bits stored or transmitted but only allow reconstruction of an approximation of the original data.

continued on next page

© 2023 College Board

LEARNING OBJECTIVE

DAT-1.D

Compare data compression algorithms to determine which is best in a particular context. 1.D

ESSENTIAL KNOWLEDGE

DAT-1.D.6

Lossy data compression algorithms can usually reduce the number of bits stored or transmitted more than lossless compression algorithms.

DAT-1.D.7

In situations where quality or ability to reconstruct the original is maximally important, lossless compression algorithms are typically chosen.

DAT-1.D.8

In situations where minimizing data size or transmission time is maximally important, lossy compression algorithms are typically chosen.

© 2023 College Board

TOPIC 2.3
Extracting Information from Data

SKILLS

5.B
Explain how knowledge can be generated from data.

5.D
Describe the impact of gathering data.

Required Course Content

ENDURING UNDERSTANDING

DAT-2
Programs can be used to process data, which allows users to discover information and create new knowledge.

LEARNING OBJECTIVE

DAT-2.A
Describe what information can be extracted from data.
5.B

ESSENTIAL KNOWLEDGE

DAT-2.A.1
Information is the collection of facts and patterns extracted from data.

DAT-2.A.2
Data provide opportunities for identifying trends, making connections, and addressing problems.

DAT-2.A.3
Digitally processed data may show correlation between variables. A correlation found in data does not necessarily indicate that a causal relationship exists. Additional research is needed to understand the exact nature of the relationship.

DAT-2.A.4
Often, a single source does not contain the data needed to draw a conclusion. It may be necessary to combine data from a variety of sources to formulate a conclusion.

continued on next page

© 2023 College Board

LEARNING OBJECTIVE	ESSENTIAL KNOWLEDGE
DAT-2.B Describe what information can be extracted from metadata. 5.B	**DAT-2.B.1** *Metadata* are data about data. For example, the piece of *data* may be an image, while the *metadata* may include the date of creation or the file size of the image. **DAT-2.B.2** Changes and deletions made to metadata do not change the primary data. **DAT-2.B.3** Metadata are used for finding, organizing, and managing information. **DAT-2.B.4** Metadata can increase the effective use of data or data sets by providing additional information. **DAT-2.B.5** Metadata allow data to be structured and organized.
DAT-2.C Identify the challenges associated with processing data. 5.D	**DAT-2.C.1** The ability to process data depends on the capabilities of the users and their tools. **DAT-2.C.2** Data sets pose challenges regardless of size, such as: ▪ the need to clean data ▪ incomplete data ▪ invalid data ▪ the need to combine data sources **DAT-2.C.3** Depending on how data were collected, they may not be uniform. For example, if users enter data into an open field, the way they choose to abbreviate, spell, or capitalize something may vary from user to user. **DAT-2.C.4** *Cleaning data* is a process that makes the data uniform without changing their meaning (e.g., replacing all equivalent abbreviations, spellings, and capitalizations with the same word). **DAT-2.C.5** Problems of bias are often created by the type or source of data being collected. Bias is not eliminated by simply collecting more data.

continued on next page

© 2023 College Board

LEARNING OBJECTIVE

DAT-2.C

Identify the challenges associated with processing data. 5.D

ESSENTIAL KNOWLEDGE

DAT-2.C.6

The size of a data set affects the amount of information that can be extracted from it.

DAT-2.C.7

Large data sets are difficult to process using a single computer and may require parallel systems.

DAT-2.C.8

Scalability of systems is an important consideration when working with data sets, as the computational capacity of a system affects how data sets can be processed and stored.

© 2023 College Board

SKILLS

2.B
Implement and apply an algorithm.

5.B
Explain how knowledge can be generated from data.

TOPIC 2.4
Using Programs with Data

Required Course Content

ENDURING UNDERSTANDING

DAT-2

Programs can be used to process data, which allows users to discover information and create new knowledge.

LEARNING OBJECTIVE

DAT-2.D

Extract information from data using a program. 2.B

ESSENTIAL KNOWLEDGE

DAT-2.D.1

Programs can be used to process data to acquire information.

DAT-2.D.2

Tables, diagrams, text, and other visual tools can be used to communicate insight and knowledge gained from data.

DAT-2.D.3

Search tools are useful for efficiently finding information.

DAT-2.D.4

Data filtering systems are important tools for finding information and recognizing patterns in data.

DAT-2.D.5

Programs such as spreadsheets help efficiently organize and find trends in information.

continued on next page

© 2023 College Board

LEARNING OBJECTIVE	ESSENTIAL KNOWLEDGE
DAT-2.D Extract information from data using a program. 2.B	**DAT-2.D.6** Some processes that can be used to extract or modify information from data include the following: ▪ transforming every element of a data set, such as doubling every element in a list, or adding a parent's email to every student record ▪ filtering a data set, such as keeping only the positive numbers from a list, or keeping only students who signed up for band from a record of all the students ▪ combining or comparing data in some way, such as adding up a list of numbers, or finding the student who has the highest GPA ▪ visualizing a data set through a chart, graph, or other visual representation
DAT-2.E Explain how programs can be used to gain insight and knowledge from data. 5.B	**DAT-2.E.1** Programs are used in an iterative and interactive way when processing information to allow users to gain insight and knowledge about data. **DAT-2.E.2** Programmers can use programs to filter and clean digital data, thereby gaining insight and knowledge. **DAT-2.E.3** Combining data sources, clustering data, and classifying data are parts of the process of using programs to gain insight and knowledge from data. **DAT-2.E.4** Insight and knowledge can be obtained from translating and transforming digitally represented information. **DAT-2.E.5** Patterns can emerge when data are transformed using programs.

© 2023 College Board

THIS PAGE IS INTENTIONALLY LEFT BLANK.

AP COMPUTER SCIENCE PRINCIPLES

BIG IDEA 3

Algorithms and Programming

30–35%
AP EXAM WEIGHTING

© 2023 College Board

Remember to go to **AP Classroom** to assign students the online Topic Questions for this big idea.

Whether assigned as homework or completed in class, the Topic Questions can be used to spot-check student understanding and help identify content and skills to emphasize in lessons.

Topic Questions

Multiple-choice: ~90 questions
Performance Tasks: ~20 prompts

© 2023 College Board

BIG IDEA 3

30–35% AP EXAM WEIGHTING

Algorithms and Programming

ESSENTIAL QUESTIONS

AAP-1

- How can we store data in a program to solve problems?

AAP-2

- What might happen if you completed the steps in your regular morning routine to get ready and go to school in a different order? How might the reordering affect the decisions you make each morning?

AAP-3

- How do video games group the different actions for a player based on what key is pressed on the keyboard or controller? How do apps group different actions together based on user interaction, such as pressing buttons?

AAP-4

- What types of problems can be solved more easily with a computer, and what types can be solved more easily without a computer? Why?

Developing Understanding

All programming languages, whether block-based or text-based, use similar programming structures and commands. Having a basic understanding of how these building blocks are combined to form algorithms and abstractions in one language makes it easier to apply these same understandings to other programming languages. This big idea focuses on determining the efficiency of algorithms, as well as writing and implementing algorithms in a program. This big idea can be paired with any of the other big ideas and taught throughout the school year.

Building Computational Thinking Practices

1.A 2.A 2.B 3.B 4.B

Some problems are so simple that writing a program would be more time-consuming than accomplishing the task by hand, while others are so complicated that it would take a computer an unreasonable amount of time to solve, if it even is possible. It is important for students to understand that not all problems can or should be solved using a program.

In procedural abstraction, programmers group program code into a procedure to make the code more readable and reusable. This type of abstraction often means using parameters to make the procedure more general and allow it to be called multiple times with varying inputs. One way to help students create abstractions is to have them write multiple versions of a procedure that each accomplish a specific instance of a task. After identifying the commonalities, students may find it easier to write a more abstract, general procedure that uses parameters to accommodate the differences.

In data abstraction, programmers use a list as a representation of something else, such as a grocery list or seating chart. When students explain how their abstractions manage complexity, they should use the context of their specific program to support why the abstraction is necessary or helpful, rather than writing in general about how abstraction manages program complexity. One way to do this is to consider how the program would be written differently, or if it could be written at all, without the use of the abstraction.

Preparing for the AP Exam

On the AP Exam, students will be asked to determine the result or functionality of code. Students can practice analyzing code segments by first predicting the result of provided code and then checking their hypothesis by running the program on a computer.

Because the Create performance task requires students to create their own program, they will need scaffolded practice that moves them from representing algorithmic processes through diagrams or pseudocode, to writing simple programs that perform calculations, to more complex programs that leverage abstraction. While students can collaborate on the development of the program, each student should be a contributing member of the program development. When completing the written responses, during the end-of-course AP Exam, an in-depth understanding of how the code segments function will allow each student to answer the prompts on their own.

© 2023 College Board

BIG IDEA AT A GLANCE

Learning Objectives	Topic	Skills	Unit/Module
AAP-1-A, AAP-1.B	**3.1 Variables and Assignments**	3.A Generalize data sources through variables. 4.B Determine the result of code segments.	
AAP-1.C, AAP-1.D	**3.2 Data Abstraction**	3.A Generalize data sources through variables. 3.B Use abstraction to manage complexity in a program. 3.C Explain how abstraction manages complexity.	
AAP-2.A, AAP-2.B, AAP-2.C	**3.3 Mathematical Expressions**	2.A Represent algorithmic processes without using a programming language. 2.B Implement and apply an algorithm. 4.B Determine the result of code segments.	
AAP-2.D	**3.4 Strings**	4.B Determine the result of code segments.	
AAP-2.E, AAP-2.F	**3.5 Boolean Expressions**	2.B Implement and apply an algorithm. 4.B Determine the result of code segments.	
AAP-2.G, AAP-2.H	**3.6 Conditionals**	2.A Represent algorithmic processes without using a programming language. 2.B Implement and apply an algorithm. 4.B Determine the result of code segments.	
AAP-2.I	**3.7 Nested Conditionals**	2.B Implement and apply an algorithm. 4.B Determine the result of code segments.	
AAP-2.J, AAP-2.K	**3.8 Iteration**	2.A Represent algorithmic processes without using a programming language. 2.B Implement and apply an algorithm. 4.B Determine the result of code segments.	
AAP-2.L, AAP-2.M	**3.9 Developing Algorithms**	1.D Evaluate solution options. 2.A Represent algorithmic processes without using a programming language. 2.B Implement and apply an algorithm.	

continued on next page

© 2023 College Board

BIG IDEA AT A GLANCE *(cont'd)*

Learning Objectives	Topic	Skills	Unit/Module
AAP-2.N, AAP-2.O	**3.10 Lists**	2.B Implement and apply an algorithm. 4.B Determine the result of code segments.	
AAP-2.P	**3.11 Binary Search**	1.A Investigate the situation, context, or task. 1.D Evaluate solution options.	
AAP-3.A	**3.12 Calling Procedures**	3.B Use abstraction to manage complexity in a program. 4.B Determine the result of code segments.	
AAP-3.B, AAP-3.C	**3.13 Developing Procedures**	3.B Use abstraction to manage complexity in a program. 3.C Explain how abstraction manages complexity.	
AAP-3.D	**3.14 Libraries**	2.B Implement and apply an algorithm.	
AAP-3.E	**3.15 Random Values**	2.B Implement and apply an algorithm. 4.B Determine the result of code segments.	
AAP-3.F	**3.16 Simulations**	1.A Investigate the situation, context, or task. 1.D Evaluate solution options.	
AAP-4.A	**3.17 Algorithmic Efficiency**	1.D Evaluate solution options.	
AAP-4.B	**3.18 Undecidable Problems**	1.A Investigate the situation, context, or task.	

Go to **AP Classroom** to assign Topic Questions as you teach the topics in Big Idea 3. Review the results in class to identify and address any student misunderstandings.

© 2023 College Board

SAMPLE INSTRUCTIONAL ACTIVITIES

The sample activities on this page are optional and are offered to provide possible ways to incorporate instructional approaches into the classroom. They were developed in partnership with teachers from the AP community to share ways that they approach teaching some of the topics and skills in this big idea. Please refer to the Instructional Strategies section beginning on p. 132 for more examples of activities and strategies.

Activity	Topic	Sample Activity
1	3.1	**Predict and compare** Provide students with a list of expressions with assignments. Ask them to predict the value of each variable after the assignment and then compare their answers to the output produced when these statements are put into a program.
2	3.6	**Using manipulatives** When learning about conditionals, take a printout of a simple conditional statement and cut it into multiple sections. As students enter the classroom, hand them an envelope full of the paper strips, and ask them to reassemble the conditional in the proper order.
3	3.12	**Marking the text** Provide students with program code that draws a square of side length 10 and a separate set of program code that uses side length 100. Ask students to mark up the sets of code to identify where they are different and to create a generalization by using parameters. Ask them to write a procedure that uses parameters to draw a square of any size.
4	3.18	**Think-pair-share** Have students work in pairs to consider what factors would be the most important to prioritize in writing an algorithm to build the perfect master schedule for the school. Some considerations may include maximum class size, student preferences, and teacher availability. Have the pairs discuss and then report their results. Finally, discuss as a class how such programs may have to settle for a "good enough" solution when an exact solution may not be possible in a reasonable amount of time.

Big Idea Planning Notes

Use the space below to plan your approach to the topics in this big idea. Consider what resources and instructional strategies you might want to use.

© 2023 College Board

BIG IDEA 3

TOPIC 3.1
Variables and Assignments

SKILLS

3.A
Generalize data sources through variables.

4.B
Determine the result of code segments.

AVAILABLE RESOURCE

- AP CSP Exam Reference Sheet (see Appendix)

Required Course Content

ENDURING UNDERSTANDING

AAP-1

To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways.

LEARNING OBJECTIVE

AAP-1.A

Represent a value with a variable. 3.A

ESSENTIAL KNOWLEDGE

AAP-1.A.1

A *variable* is an abstraction inside a program that can hold a value. Each variable has associated data storage that represents one value at a time, but that value can be a list or other collection that in turn contains multiple values.

AAP-1.A.2

Using meaningful variable names helps with the readability of program code and understanding of what values are represented by the variables.

AAP-1.A.3

Some programming languages provide *types* to represent data, which are referenced using variables. These types include numbers, Booleans, lists, and strings.

AAP-1.A.4

Some values are better suited to representation using one type of datum rather than another.

continued on next page

© 2023 College Board

LEARNING OBJECTIVE

AAP-1.B

Determine the value of a variable as a result of an assignment. **4.B**

ESSENTIAL KNOWLEDGE

AAP-1.B.1

The assignment operator allows a program to change the value represented by a variable.

AAP-1.B.2

The exam reference sheet provides the "←" operator to use for assignment. For example,

Text:

```
a ← expression
```

Block:

```
a ← expression
```

evaluates `expression` and then assigns a copy of the result to the variable `a`.

AAP-1.B.3

The value stored in a variable will be the most recent value assigned. For example:

```
a ← 1
b ← a
a ← 2
display(b)
```

still displays `1`.

© 2023 College Board

BIG IDEA 3

TOPIC 3.2
Data Abstraction

SKILLS

3.A
Generalize data sources through variables.

3.B
Use abstraction to manage complexity in a program.

3.C
Explain how abstraction manages complexity.

AVAILABLE RESOURCE

- AP CSP Exam Reference Sheet (see Appendix)

Required Course Content

ENDURING UNDERSTANDING

AAP-1

To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways.

LEARNING OBJECTIVE

AAP-1.C

Represent a list or string using a variable. 3.A

ESSENTIAL KNOWLEDGE

AAP-1.C.1

A *list* is an ordered sequence of elements. For example,

```
[value1, value2, value3, ...]
```

describes a list where `value1` is the first element, `value2` is the second element, `value3` is the third element, and so on.

AAP-1.C.2

An *element* is an individual value in a list that is assigned a unique index.

AAP-1.C.3

An *index* is a common method for referencing the elements in a list or string using natural numbers.

AAP-1.C.4

A *string* is an ordered sequence of characters.

continued on next page

© 2023 College Board

LEARNING OBJECTIVE

AAP-1.D

For data abstraction:

a. Develop data abstraction using lists to store multiple elements. 3.B

b. Explain how the use of data abstraction manages complexity in program code. 3.C

ESSENTIAL KNOWLEDGE

AAP-1.D.1

Data abstraction provides a separation between the abstract properties of a data type and the concrete details of its representation.

AAP-1.D.2

Data abstractions manage complexity in programs by giving a collection of data a name without referencing the specific details of the representation.

AAP-1.D.3

Data abstractions can be created using lists.

AAP-1.D.4

Developing a data abstraction to implement in a program can result in a program that is easier to develop and maintain.

AAP-1.D.5

Data abstractions often contain different types of elements.

AAP-1.D.6

The use of lists allows multiple related items to be treated as a single value. Lists are referred to by different names, such as *array*, depending on the programming language.

X EXCLUSION STATEMENT (EK APP-1.D.6):
The use of linked lists is outside the scope of this course and the AP Exam.

AAP-1.D.7

The exam reference sheet provides the notation

```
[value1, value2, value3, ...]
```

to create a list with those values as the first, second, third, and so on items. For example,

- Text:

  ```
  aList ← [value1, value2,
  value3, ...]
  ```

  Block:

  creates a new list that contains the values `value1`, `value2`, `value3`, and `...` at indices `1`, `2`, `3`, and `...` respectively and assigns it to `aList`.

- Text:

  ```
  aList ← []
  ```

  Block:

  creates a new empty list and assigns it to `aList`.

continued on next page

© 2023 College Board

LEARNING OBJECTIVE

AAP-1.D

For data abstraction:

a. Develop data abstraction using lists to store multiple elements. 3.B

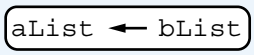
b. Explain how the use of data abstraction manages complexity in program code. 3.C

ESSENTIAL KNOWLEDGE

- Text:

  `aList ← bList`

  Block:

  assigns a copy of the list `bList` to the list `aList`. For example, if `bList` contains `[20, 40, 60]`, then `aList` will also contain `[20, 40, 60]` after the assignment.

AAP-1.D.8

The exam reference sheet describes a list structure whose index values are 1 through the number of elements in the list, inclusive. For all list operations, if a list index is less than 1 or greater than the length of the list, an error message is produced and the program will terminate.

© 2023 College Board

SKILLS

2.A
Represent algorithmic processes without using a programming language.

2.B
Implement and apply an algorithm.

4.B
Determine the result of code segments.

AVAILABLE RESOURCE

- AP CSP Exam Reference Sheet (see Appendix)

TOPIC 3.3
Mathematical Expressions

Required Course Content

ENDURING UNDERSTANDING

AAP-2

The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.

LEARNING OBJECTIVE	ESSENTIAL KNOWLEDGE
AAP-2.A Express an algorithm that uses sequencing without using a programming language. **2.A**	**AAP-2.A.1** An *algorithm* is a finite set of instructions that accomplish a specific task. **AAP-2.A.2** Beyond visual and textual programming languages, algorithms can be expressed in a variety of ways, such as natural language, diagrams, and pseudocode. **AAP-2.A.3** Algorithms executed by programs are implemented using programming languages. **AAP-2.A.4** Every algorithm can be constructed using combinations of sequencing, selection, and iteration.
AAP-2.B Represent a step-by-step algorithmic process using sequential code statements. **2.B**	**AAP-2.B.1** *Sequencing* is the application of each step of an algorithm in the order in which the code statements are given. **AAP-2.B.2** A *code statement* is a part of program code that expresses an action to be carried out.

continued on next page

© 2023 College Board

LEARNING OBJECTIVE	ESSENTIAL KNOWLEDGE
AAP-2.B Represent a step-by-step algorithmic process using sequential code statements. 2.B	**AAP-2.B.3** An *expression* can consist of a value, a variable, an operator, or a procedure call that returns a value. **AAP-2.B.4** Expressions are evaluated to produce a single value. **AAP-2.B.5** The evaluation of expressions follows a set order of operations defined by the programming language. **AAP-2.B.6** Sequential statements execute in the order they appear in the code segment. **AAP-2.B.7** Clarity and readability are important considerations when expressing an algorithm in a programming language.
AAP-2.C Evaluate expressions that use arithmetic operators. 4.B	**AAP-2.C.1** Arithmetic operators are part of most programming languages and include addition, subtraction, multiplication, division, and modulus operators. **AAP-2.C.2** The exam reference sheet provides `a MOD b`, which evaluates to the remainder when `a` is divided by `b`. Assume that `a` is an integer greater than or equal to `0` and `b` is an integer greater than `0`. For example, `17 MOD 5` evaluates to `2`. **AAP-2.C.3** The exam reference sheet provides the arithmetic operators `+`, `–`, `*`, `/`, and `MOD`. Text and Block: ▪ `a + b` ▪ `a – b` ▪ `a * b` ▪ `a / b` ▪ `a MOD b` These are used to perform arithmetic on `a` and `b`. For example, `17 / 5` evaluates to `3.4`. **AAP-2.C.4** The order of operations used in mathematics applies when evaluating expressions. The `MOD` operator has the same precedence as the `*` and `/` operators.

© 2023 College Board

SKILL

4.B

Determine the result of code segments.

TOPIC 3.4
Strings

Required Course Content

ENDURING UNDERSTANDING

AAP-2

The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.

LEARNING OBJECTIVE

AAP-2.D

Evaluate expressions that manipulate strings. 4.B

ESSENTIAL KNOWLEDGE

AAP-2.D.1

String concatenation joins together two or more strings end-to-end to make a new string.

AAP-2.D.2

A *substring* is part of an existing string.

© 2023 College Board

BIG IDEA 3

TOPIC 3.5
Boolean Expressions

SKILLS

2.B
Implement and apply an algorithm.

4.B
Determine the result of code segments.

AVAILABLE RESOURCE
- AP CSP Exam Reference Sheet (see Appendix)

Required Course Content

ENDURING UNDERSTANDING

AAP-2

The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.

LEARNING OBJECTIVE

AAP-2.E

For relationships between two variables, expressions, or values:

a. Write expressions using relational operators. 2.B

b. Evaluate expressions that use relational operators. 4.B

ESSENTIAL KNOWLEDGE

AAP-2.E.1

A *Boolean value* is either true or false.

AAP-2.E.2

The exam reference sheet provides the following relational operators: `=`, `≠`, `>`, `<`, `≥`, and `≤`.

Text and Block:

- `a = b`
- `a ≠ b`
- `a > b`
- `a < b`
- `a ≥ b`
- `a ≤ b`

These are used to test the relationship between two variables, expressions, or values. A comparison using a relational operator evaluates to a Boolean value. For example, `a = b` evaluates to `true` if `a` and `b` are equal; otherwise, it evaluates to `false`.

continued on next page

© 2023 College Board

LEARNING OBJECTIVE

AAP-2.F

For relationships between Boolean values:

a. Write expressions using logical operators. 2.B

b. Evaluate expressions that use logic operators. 4.B

ESSENTIAL KNOWLEDGE

AAP-2.F.1

The exam reference sheet provides the logical operators `NOT`, `AND`, and `OR`, which evaluate to a Boolean value.

AAP-2.F.2

The exam reference sheet provides

Text:

```
NOT condition
```

Block:

which evaluates to `true` if `condition` is `false`; otherwise it evaluates to `false`.

AAP-2.F.3

The exam reference sheet provides

Text:

```
condition1 AND condition2
```

Block:

which evaluates to `true` if both `condition1` and `condition2` are `true`; otherwise it evaluates to `false`.

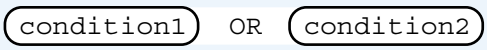
AAP-2.F.4

The exam reference sheet provides

Text:

```
condition1 OR condition2
```

Block:



which evaluates to `true` if `condition1` is `true` or if `condition2` is `true` or if both `condition1` and `condition2` are `true`; otherwise it evaluates to `false`.

AAP-2.F.5

The operand for a logical operator is either a Boolean expression or a single Boolean value.

© 2023 College Board

TOPIC 3.6
Conditionals

Required Course Content

ENDURING UNDERSTANDING

AAP-2

The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.

LEARNING OBJECTIVE

AAP-2.G

Express an algorithm that uses selection without using a programming language. 2.A

AAP-2.H

For selection:

a. Write conditional statements. 2.B

b. Determine the result of conditional statements. 4.B

ESSENTIAL KNOWLEDGE

AAP-2.G.1

Selection determines which parts of an algorithm are executed based on a condition being `true` or `false`.

AAP-2.H.1

Conditional statements, or "if-statements," affect the sequential flow of control by executing different statements based on the value of a Boolean expression.

AAP-2.H.2

The exam reference sheet provides

Text:

```
IF(condition)
{
   <block of statements>
}
```

Block:

continued on next page

SKILLS

2.A
Represent algorithmic processes without using a programming language.

2.B
Implement and apply an algorithm.

4.B
Determine the result of code segments.

AVAILABLE RESOURCES

- Professional Development > **Teaching and Assessing: Algorithmic Thinking**
- AP CSP Exam Reference Sheet (see Appendix)

© 2023 College Board

LEARNING OBJECTIVE

AAP-2.H

For selection:

a. Write conditional statements. 2.B

b. Determine the result of conditional statements. 4.B

ESSENTIAL KNOWLEDGE

in which the code in `block of statements` is executed if the Boolean expression `condition` evaluates to `true`; no action is taken if `condition` evaluates to `false`.

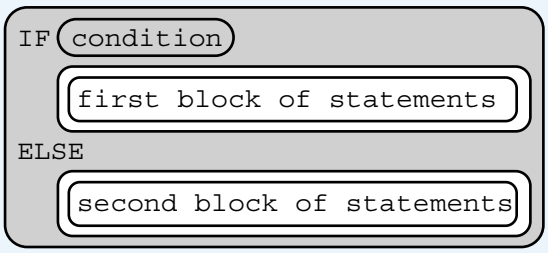
AAP-2.H.3

The exam reference sheet provides

Text:

```
IF(condition)
{
   <first block of statements>
}
ELSE
{
   <second block of statements>
}
```

Block:



in which the code in `first block of statements` is executed if the Boolean expression `condition` evaluates to `true`; otherwise, the code in `second block of statements` is executed.

© 2023 College Board

TOPIC 3.7
Nested Conditionals

SKILLS

2.B
Implement and apply an algorithm.

4.B
Determine the result of code segments.

Required Course Content

ENDURING UNDERSTANDING

AAP-2

The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.

LEARNING OBJECTIVE

AAP-2.I

For nested selection:

a. Write nested conditional statements. 2.B

b. Determine the result of nested conditional statements. 4.B

ESSENTIAL KNOWLEDGE

AAP-2.I.1

Nested conditional statements consist of conditional statements within conditional statements.

© 2023 College Board

SKILLS

2.A
Represent algorithmic processes without using a programming language.

2.B
Implement and apply an algorithm.

4.B
Determine the result of code segments.

AVAILABLE RESOURCES

- Professional Development > **Teaching and Assessing: Algorithmic Thinking**
- AP CSP Exam Reference Sheet (see Appendix)

TOPIC 3.8
Iteration

Required Course Content

ENDURING UNDERSTANDING

AAP-2

The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.

LEARNING OBJECTIVE

AAP-2.J

Express an algorithm that uses iteration without using a programming language. 2.A

AAP-2.K

For iteration:

a. Write iteration statements. 2.B

b. Determine the result or side effect of iteration statements. 4.B

ESSENTIAL KNOWLEDGE

AAP-2.J.1

Iteration is a repeating portion of an algorithm. Iteration repeats a specified number of times or until a given condition is met.

AAP-2.K.1

Iteration statements change the sequential flow of control by repeating a set of statements zero or more times, until a stopping condition is met.

AAP-2.K.2

The exam reference sheet provides

Text:

```
REPEAT n TIMES
{
   <block of statements>
}
```

Block:

in which the `block of statements` is executed `n` times.

continued on next page

© 2023 College Board

LEARNING OBJECTIVE

AAP-2.K

For iteration:

a. Write iteration statements. 2.B

b. Determine the result or side effect of iteration statements. 4.B

ESSENTIAL KNOWLEDGE

AAP-2.K.3

The exam reference sheet provides

Text:

```
REPEAT UNTIL(condition)
{
   <block of statements>
}
```

Block:

in which the code in `block of statements` is repeated until the Boolean expression `condition` evaluates to `true`.

AAP-2.K.4

In `REPEAT UNTIL(condition)` iteration, an infinite loop occurs when the ending condition will never evaluate to `true`.

AAP-2.K.5

In `REPEAT UNTIL(condition)` iteration, if the conditional evaluates to `true` initially, the loop body is not executed at all, due to the condition being checked before the loop.

© 2023 College Board

SKILLS

1.D
Evaluate solution options.

2.A
Represent algorithmic processes without using a programming language.

2.B
Implement and apply an algorithm.

TOPIC 3.9
Developing Algorithms

Required Course Content

ENDURING UNDERSTANDING

AAP-2

The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.

LEARNING OBJECTIVE

AAP-2.L

Compare multiple algorithms to determine if they yield the same side effect or result. **1.D**

ESSENTIAL KNOWLEDGE

AAP-2.L.1
Algorithms can be written in different ways and still accomplish the same tasks.

AAP-2.L.2
Algorithms that appear similar can yield different side effects or results.

AAP-2.L.3
Some conditional statements can be written as equivalent Boolean expressions.

AAP-2.L.4
Some Boolean expressions can be written as equivalent conditional statements.

AAP-2.L.5
Different algorithms can be developed or used to solve the same problem.

continued on next page

© 2023 College Board

Algorithms and Programming

LEARNING OBJECTIVE

AAP-2.M

For algorithms:

a. Create algorithms. 2.A

b. Combine and modify existing algorithms. 2.B

ESSENTIAL KNOWLEDGE

AAP-2.M.1

Algorithms can be created from an idea, by combining existing algorithms, or by modifying existing algorithms.

AAP-2.M.2

Knowledge of existing algorithms can help in constructing new ones. Some existing algorithms include:

- determining the maximum or minimum value of two or more numbers
- computing the sum or average of two or more numbers
- identifying if an integer is or is not evenly divisible by another integer
- determining a robot's path through a maze

AAP-2.M.3

Using existing correct algorithms as building blocks for constructing another algorithm has benefits such as reducing development time, reducing testing, and simplifying the identification of errors.

© 2023 College Board

SKILLS

2.B
Implement and apply an algorithm.

4.B
Determine the result of code segments.

AVAILABLE RESOURCE

- AP CSP Exam Reference Sheet (see Appendix)

TOPIC 3.10
Lists

Required Course Content

ENDURING UNDERSTANDING

AAP-2

The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.

LEARNING OBJECTIVE

AAP-2.N

For list operations:

a. Write expressions that use list indexing and list procedures. 2.B

b. Evaluate expressions that use list indexing and list procedures. 4.B

ESSENTIAL KNOWLEDGE

AAP-2.N.1

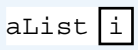
The exam reference sheet provides basic operations on lists, including:

- accessing an element by index

  Text:

  `aList[i]`

  Block:

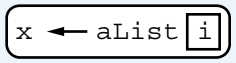
  

  accesses the element of `aList` at index `i`. The first element of `aList` is at index `1` and is accessed using the notation `aList[1]`.
- assigning a value of an element of a list to a variable

  Text:

  `x ← aList [i]`

  Block:

  assigns the value of `aList[i]` to the variable `x`.
- assigning a value to an element of a list

  Text:

  `aList[i] ← x`

  Block:

  assigns the value of `x` to `aList[i]`.

continued on next page

© 2023 College Board

LEARNING OBJECTIVE

AAP-2.N

For list operations:

a. Write expressions that use list indexing and list procedures. 2.B

b. Evaluate expressions that use list indexing and list procedures. 4.B

ESSENTIAL KNOWLEDGE

Text:

```
aList[i] ← aList[j]
```

Block:

assigns the value of `aList[j]` to `aList[i]`.

- inserting elements at a given index

  Text:

  ```
  INSERT(aList, i, value)
  ```

  Block:

  shifts to the right any values in `aList` at indices greater than or equal to `i`. The length of the list is increased by 1, and `value` is placed at index `i` in `aList`.

- adding elements to the end of the list

  Text:

  ```
  APPEND(aList, value)
  ```

  Block:

  increases the length of `aList` by 1, and `value` is placed at the end of `aList`.

- removing elements

  Text:

  ```
  REMOVE(aList, i)
  ```

  Block:

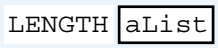
  removes the item at index `i` in `aList` and shifts to the left any values at indices greater than `i`. The length of `aList` is decreased by 1.

- determining the length of a list

  Text:

  ```
  LENGTH(aList)
  ```

  Block:

  evaluates to the number of elements currently in `aList`.

AAP-2.N.2

List procedures are implemented in accordance with the syntax rules of the programming language.

continued on next page

© 2023 College Board

LEARNING OBJECTIVE

AAP-2.O

For algorithms involving elements of a list:

a. Write iteration statements to traverse a list. 2.B

b. Determine the result of an algorithm that includes list traversals. 4.B

ESSENTIAL KNOWLEDGE

AAP-2.O.1

Traversing a list can be a complete traversal, where all elements in the list are accessed, or a partial traversal, where only a portion of elements are accessed.

X EXCLUSION STATEMENT (EK AAP-2.O.1):

Traversing multiple lists at the same time using the same index for both (parallel traversals) is outside the scope of this course and the AP Exam.

AAP-2.O.2

Iteration statements can be used to traverse a list.

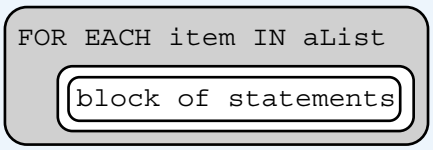
AAP-2.O.3

The exam reference sheet provides

Text:

```
FOR EACH item IN aList
{
 <block of statements>
}
```

Block:



The variable `item` is assigned the value of each element of `aList` sequentially, in order, from the first element to the last element. The code in `block of statements` is executed once for each assignment of `item`.

AAP-2.O.4

Knowledge of existing algorithms that use iteration can help in constructing new algorithms. Some examples of existing algorithms that are often used with lists include:

- determining a minimum or maximum value in a list
- computing a sum or average of a list of numbers

AAP-2.O.5

Linear search or sequential search algorithms check each element of a list, in order, until the desired value is found or all elements in the list have been checked.

© 2023 College Board

TOPIC 3.11
Binary Search

SKILLS

1.A
Investigate the situation, context, or task.

1.D
Evaluate solution options.

AVAILABLE RESOURCES

- Classroom Resources > **Binary Search**
- External Resource > **Searching Algorithms** from CS Unplugged

Required Course Content

ENDURING UNDERSTANDING

AAP-2

The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.

LEARNING OBJECTIVE

AAP-2.P

For binary search algorithms:

a. Determine the number of iterations required to find a value in a data set. 1.D

b. Explain the requirements necessary to complete a binary search. 1.A

ESSENTIAL KNOWLEDGE

AAP-2.P.1

The binary search algorithm starts at the middle of a sorted data set of numbers and eliminates half of the data; this process repeats until the desired value is found or all elements have been eliminated.

EXCLUSION STATEMENT (EK: AAP-2.P.1):
Specific implementations of the binary search are outside the scope of the course and the AP Exam.

AAP-2.P.2

Data must be in sorted order to use the binary search algorithm.

AAP-2.P.3

Binary search is often more efficient than sequential/linear search when applied to sorted data.

© 2023 College Board

SKILLS

3.B
Use abstraction to manage complexity in a program.

4.B
Determine the result of code segments.

AVAILABLE RESOURCE

- AP CSP Exam Reference Sheet (see Appendix)

TOPIC 3.12
Calling Procedures

Required Course Content

ENDURING UNDERSTANDING

AAP-3

Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing them to write programs more quickly and with more confidence.

LEARNING OBJECTIVE

AAP-3.A

For procedure calls:

a. Write statements to call procedures. 3.B

b. Determine the result or effect of a procedure call. 4.B

ESSENTIAL KNOWLEDGE

AAP-3.A.1

A *procedure* is a named group of programming instructions that may have parameters and return values.

AAP-3.A.2

Procedures are referred to by different names, such as *method* or *function*, depending on the programming language.

AAP-3.A.3

Parameters are input variables of a procedure. *Arguments* specify the values of the parameters when a procedure is called.

AAP-3.A.4

A procedure call interrupts the sequential execution of statements, causing the program to execute the statements within the procedure before continuing. Once the last statement in the procedure (or a return statement) has executed, flow of control is returned to the point immediately following where the procedure was called.

AAP-3.A.5

The exam reference sheet provides

```
procName(arg1, arg2, ...)
```

continued on next page

© 2023 College Board

LEARNING OBJECTIVE

AAP-3.A

For procedure calls:

a. Write statements to call procedures. 3.B

b. Determine the result or effect of a procedure call. 4.B

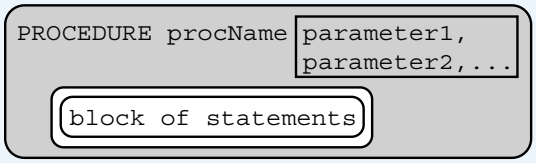
ESSENTIAL KNOWLEDGE

as a way to call

Text:

```
PROCEDURE procName(parameter1,
                   parameter2, ...)
{
    <block of statements>
}
```

Block:



which takes zero or more arguments; `arg1` is assigned to `parameter1`, `arg2` is assigned to `parameter2`, and so on.

AAP-3.A.6

The exam reference sheet provides the procedure

Text:

```
DISPLAY(expression)
```

Block:

to display the value of `expression`, followed by a space.

AAP-3.A.7

The exam reference sheet provides the

Text:

```
RETURN(expression)
```

Block:

statement, which is used to return the flow of control to the point where the procedure was called and to return the value of `expression`.

AAP-3.A.8

The exam reference sheet provides

```
result ← procName(arg1,
                  arg2, ...)
```

to assign to `result` the "value of the procedure" being returned by calling

continued on next page

© 2023 College Board

LEARNING OBJECTIVE

AAP-3.A

For procedure calls:

a. Write statements to call procedures. 3.B

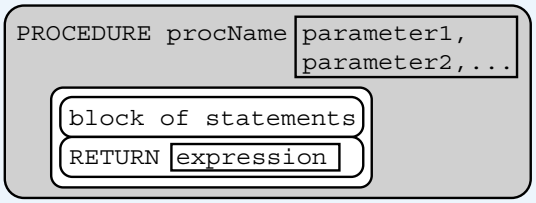
b. Determine the result or effect of a procedure call. 4.B

ESSENTIAL KNOWLEDGE

Text:

```
PROCEDURE procName(parameter1,
                   parameter2, ...)
{
   <block of statements>
   RETURN(expression)
}
```

Block:



AAP-3.A.9

The exam reference sheet provides procedure

Text:

```
INPUT()
```

Block:

```
INPUT
```

which accepts a value from the user and returns the input value.

© 2023 College Board

BIG IDEA 3

TOPIC 3.13
Developing Procedures

SKILLS

3.B
Use abstraction to manage complexity in a program.

3.C
Explain how abstraction manages complexity.

AVAILABLE RESOURCE
- AP CSP Exam Reference Sheet (see Appendix)

Required Course Content

ENDURING UNDERSTANDING

AAP-3

Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing them to write programs more quickly and with more confidence.

LEARNING OBJECTIVE

AAP-3.B
Explain how the use of procedural abstraction manages complexity in a program. 3.C

ESSENTIAL KNOWLEDGE

AAP-3.B.1
One common type of abstraction is procedural abstraction, which provides a name for a process and allows a procedure to be used only knowing what it does, not how it does it.

AAP-3.B.2
Procedural abstraction allows a solution to a large problem to be based on the solutions of smaller subproblems. This is accomplished by creating procedures to solve each of the subproblems.

AAP-3.B.3
The subdivision of a computer program into separate subprograms is called *modularity*.

AAP-3.B.4
A procedural abstraction may extract shared features to generalize functionality instead of duplicating code. This allows for program code reuse, which helps manage complexity.

AAP-3.B.5
Using parameters allows procedures to be generalized, enabling the procedures to be reused with a range of input values or arguments.

continued on next page

© 2023 College Board

LEARNING OBJECTIVE

AAP-3.B

Explain how the use of procedural abstraction manages complexity in a program. 3.C

ESSENTIAL KNOWLEDGE

AAP-3.B.6

Using procedural abstraction helps improve code readability.

AAP-3.B.7

Using procedural abstraction in a program allows programmers to change the internals of the procedure (to make it faster, more efficient, use less storage, etc.) without needing to notify users of the change as long as what the procedure does is preserved.

AAP-3.C

Develop procedural abstractions to manage complexity in a program by writing procedures. 3.B

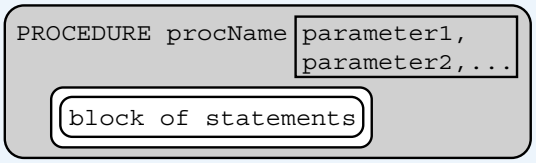
AAP-3.C.1

The exam reference sheet provides

Text:

```
PROCEDURE procName(parameter1,
                   parameter2, ...)
{
   <block of statements>
}
```

Block:



which is used to define a procedure that takes zero or more arguments. The procedure contains `block of statements`.

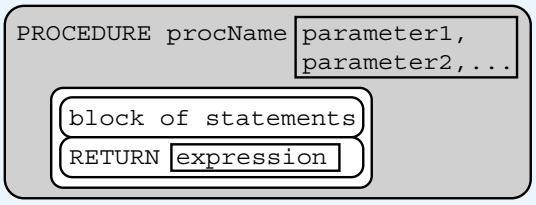
AAP-3.C.2

The exam reference sheet provides

Text:

```
PROCEDURE procName(parameter1,
                   parameter2, ...)
{
   <block of statements>
   RETURN(expression)
}
```

Block:



continued on next page

© 2023 College Board

LEARNING OBJECTIVE	ESSENTIAL KNOWLEDGE
AAP-3.C Develop procedural abstractions to manage complexity in a program by writing procedures. 3.B	which is used to define a procedure that takes zero or more arguments. The procedure contains `block of statements` and returns the value of `expression`. The `RETURN` statement may appear at any point inside the procedure and causes an immediate return from the procedure back to the calling statement.

© 2023 College Board

SKILL

2.B

Implement and apply an algorithm.

TOPIC 3.14
Libraries

Required Course Content

ENDURING UNDERSTANDING

AAP-3

Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing them to write programs more quickly and with more confidence.

LEARNING OBJECTIVE

AAP-3.D

Select appropriate libraries or existing code segments to use in creating new programs. 2.B

ESSENTIAL KNOWLEDGE

AAP-3.D.1

A software library contains procedures that may be used in creating new programs.

AAP-3.D.2

Existing code segments can come from internal or external sources, such as libraries or previously written code.

AAP-3.D.3

The use of libraries simplifies the task of creating complex programs.

AAP-3.D.4

Application program interfaces (APIs) are specifications for how the procedures in a library behave and can be used.

AAP-3.D.5

Documentation for an API/library is necessary in understanding the behaviors provided by the API/library and how to use them.

© 2023 College Board

BIG IDEA 3

TOPIC 3.15
Random Values

SKILLS

2.B
Implement and apply an algorithm.

4.B
Determine the result of code segments.

AVAILABLE RESOURCES

- Professional Development > **Teaching and Assessing Module: Algorithmic Thinking**
- AP CSP Exam Reference Sheet (see Appendix)

Required Course Content

ENDURING UNDERSTANDING

AAP-3

Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing them to write programs more quickly and with more confidence.

LEARNING OBJECTIVE

AAP-3.E

For generating random values:

a. Write expressions to generate possible values. 2.B

b. Evaluate expressions to determine the possible results. 4.B

ESSENTIAL KNOWLEDGE

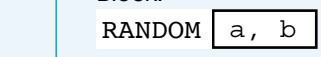
AAP-3.E.1

The exam reference sheet provides

Text:

```
RANDOM(a, b)
```

Block:



which generates and returns a random integer from `a` to `b`, inclusive. Each result is equally likely to occur. For example, `RANDOM(1, 3)` could return `1`, `2`, or `3`.

AAP-3.E.2

Using random number generation in a program means each execution may produce a different result.

© 2023 College Board

SKILLS

1.A
Investigate the situation, context, or task.

1.D
Evaluate solution options.

TOPIC 3.16
Simulations

Required Course Content

ENDURING UNDERSTANDING

AAP-3

Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing them to write programs more quickly and with more confidence.

LEARNING OBJECTIVE

AAP-3.F

For simulations:

a. Explain how computers can be used to represent real-world phenomena or outcomes. 1.A

b. Compare simulations with real-world contexts. 1.D

ESSENTIAL KNOWLEDGE

AAP-3.F.1
Simulations are abstractions of more complex objects or phenomena for a specific purpose.

AAP-3.F.2
A *simulation* is a representation that uses varying sets of values to reflect the changing state of a phenomenon.

AAP-3.F.3
Simulations often mimic real-world events with the purpose of drawing inferences, allowing investigation of a phenomenon without the constraints of the real world.

AAP-3.F.4
The process of developing an abstract simulation involves removing specific details or simplifying functionality.

AAP-3.F.5
Simulations can contain bias derived from the choices of real-world elements that were included or excluded.

AAP-3.F.6
Simulations are most useful when real-world events are impractical for experiments (e.g., too big, too small, too fast, too slow, too expensive, or too dangerous).

continued on next page

© 2023 College Board

LEARNING OBJECTIVE

AAP-3.F

For simulations:

a. Explain how computers can be used to represent real-world phenomena or outcomes. 1.A

b. Compare simulations with real-world contexts. 1.D

ESSENTIAL KNOWLEDGE

AAP-3.F.7

Simulations facilitate the formulation and refinement of hypotheses related to the objects or phenomena under consideration.

AAP-3.F.8

Random number generators can be used to simulate the variability that exists in the real world.

© 2023 College Board

SKILL
1.D
Evaluate solution options.

TOPIC 3.17
Algorithmic Efficiency

Required Course Content

ENDURING UNDERSTANDING

AAP-4

There exist problems that computers cannot solve, and even when a computer can solve a problem, it may not be able to do so in a reasonable amount of time.

LEARNING OBJECTIVE

AAP-4.A

For determining the efficiency of an algorithm:

a. Explain the difference between algorithms that run in reasonable time and those that do not. 1.D

b. Identify situations where a heuristic solution may be more appropriate. 1.D

ESSENTIAL KNOWLEDGE

AAP-4.A.1

A *problem* is a general description of a task that can (or cannot) be solved algorithmically. An *instance* of a problem also includes specific input. For example, sorting is a problem; sorting the list (2,3,1,7) is an instance of the problem.

AAP-4.A.2

A *decision problem* is a problem with a yes/no answer (e.g., is there a path from A to B?). An *optimization problem* is a problem with the goal of finding the "best" solution among many (e.g., what is the shortest path from A to B?).

AAP-4.A.3

Efficiency is an estimation of the amount of computational resources used by an algorithm. Efficiency is typically expressed as a function of the size of the input.

EXCLUSION STATEMENT (EK AAP-4.A.3): *Formal analysis of algorithms (Big-O) and formal reasoning using mathematical formulas are outside the scope of this course and the AP Exam.*

AAP-4.A.4

An algorithm's efficiency is determined through formal or mathematical reasoning.

continued on next page

© 2023 College Board

LEARNING OBJECTIVE

AAP-4.A

For determining the efficiency of an algorithm:

a. Explain the difference between algorithms that run in reasonable time and those that do not. 1.D

b. Identify situations where a heuristic solution may be more appropriate. 1.D

ESSENTIAL KNOWLEDGE

AAP-4.A.5

An algorithm's efficiency can be informally measured by determining the number of times a statement or group of statements executes.

AAP-4.A.6

Different correct algorithms for the same problem can have different efficiencies.

AAP-4.A.7

Algorithms with a polynomial efficiency or slower (constant, linear, square, cube, etc.) are said to run in a *reasonable amount of time*. Algorithms with exponential or factorial efficiencies are examples of algorithms that run in an *unreasonable amount of time*.

AAP-4.A.8

Some problems cannot be solved in a reasonable amount of time because there is no efficient algorithm for solving them. In these cases, approximate solutions are sought.

AAP-4.A.9

A *heuristic* is an approach to a problem that produces a solution that is not guaranteed to be optimal but may be used when techniques that are guaranteed to always find an optimal solution are impractical.

EXCLUSION STATEMENT (AAP-4.A.9):
Specific heuristic solutions are outside the scope of this course and the AP Exam.

© 2023 College Board

SKILL

1.A

Investigate the situation, context, or task.

TOPIC 3.18
Undecidable Problems

Required Course Content

ENDURING UNDERSTANDING

AAP-4

There exist problems that computers cannot solve, and even when a computer can solve a problem, it may not be able to do so in a reasonable amount of time.

LEARNING OBJECTIVE

AAP-4.B

Explain the existence of undecidable problems in computer science. 1.A

ESSENTIAL KNOWLEDGE

AAP-4.B.1

A *decidable problem* is a decision problem for which an algorithm can be written to produce a correct output for all inputs (e.g., "Is the number even?").

AAP-4.B.2

An *undecidable problem* is one for which no algorithm can be constructed that is always capable of providing a correct yes-or-no answer.

EXCLUSION STATEMENT (EK AAP-4.B.2): *Determining whether a given problem is undecidable is outside the scope of this course and the AP Exam.*

AAP-4.B.3

An undecidable problem may have some instances that have an algorithmic solution, but there is no algorithmic solution that could solve all instances of the problem.

© 2023 College Board

AP COMPUTER SCIENCE PRINCIPLES

BIG IDEA 4

Computer Systems and Networks

11–15%
AP EXAM WEIGHTING

© 2023 College Board

Remember to go to **AP Classroom** to assign students the online Topic Questions for this big idea.

Whether assigned as homework or completed in class, the Topic Questions can be used to spot-check student understanding and help identify content and skills to emphasize in lessons.

Topic Questions

Multiple-choice: ~10 questions

© 2023 College Board

BIG IDEA 4

11–15% AP EXAM WEIGHTING

Computer Systems and Networks

ESSENTIAL QUESTIONS

CSN-1

- Why are long text messages sometime delivered out of order?
- When an Internet service outage occurs in a different part of your town or city, how are you still able to access the Internet?

CSN-2

- What are the benefits of dividing tasks among group members?
- Is there a point where adding another group member would not make completing the task faster? Why?

Developing Understanding

The Internet is a network that most students use on a regular basis to look up information, to socialize with friends, and in many cases to complete their school work. In this big idea, students will learn how computer systems and networks, primarily the Internet, work. Students will learn about how information is transmitted on the Internet and about the safeguards that have been put in place to keep this system from breaking down. In addition, students will learn the effect that dividing tasks across multiple computing devices can have on the speed at which processes can occur. This big idea is often taught in conjunction with Big Idea 5: Impact of Computing.

Building Computational Thinking Practices

1.D 5.A

Some programs are so large or complex that they end up taking a long time to run, making them impractical to use. In order to minimize this runtime, the individual processes that comprise a program can be run on multiple processors simultaneously. Programmers need to evaluate the options of running these processes sequentially or in parallel across multiple processors to optimize the solution time. When introducing students to distributed processes, use real-world examples, such as dividing up household chores or passing back classroom papers.

To create more stable access to the Internet, additional connections and routers can be added. Redundant routing options help ensure that the Internet is more reliable and stable. To determine where additional connections and routers need to be added to the systems, evaluation of existing infrastructure is beneficial. Using kinesthetic learning techniques to simulate the connections and routers in a system can help students make this abstract idea more concrete. As student understanding increases, propose problems for the students to solve, such as broken connections or the need to expand the system to include newly added routers or devices.

Preparing for the AP Exam

On the end-of-course exam, students will be presented with scenarios for how information could be passed via the Internet along with illustrations of interconnected computers in given networks. Students will be asked to select which choice best explains how information is passed through these networks from one computing device to another or how designing systems to include redundancy helps make them fault-tolerant. To aid in understanding these diagrams, students should be encouraged to use the Marking the Text strategy (see page 133). They can mark the diagrams with the information provided to ensure they have an accurate visual and can then match that visual to the explanations provided.

© 2023 College Board

BIG IDEA AT A GLANCE

Learning Objective	Topic	Skills	Unit/Module
CSN-1.A, CSN-1.B, CSN-1.C, CSN-1.D	**4.1** The Internet	5.A Explain how computing systems work.	
CSN-1.E	**4.2** Fault Tolerance	1.D Evaluate solution options. 5.A Explain how computing systems work.	
CSN-2.A, CSN-2.B	**4.3** Parallel and Distributed Computing	1.D Evaluate solution options.	

Go to **AP Classroom** to assign Topic Questions as you teach the topics in Big Idea 4. Review the results in class to identify and address any student misunderstandings.

© 2023 College Board

SAMPLE INSTRUCTIONAL ACTIVITIES

The sample activities on this page are optional and are offered to provide possible ways to incorporate instructional approaches into the classroom. They were developed in partnership with teachers from the AP community to share ways that they approach teaching some of the topics and skills in this big idea. Please refer to the Instructional Strategies section beginning on p. 132 for more examples of activities and strategies.

Activity	Topic	Sample Activity
1	4.1	**Journaling** Ask students to read about the Internet and packet switching in **Blown to Bits**. Pose several prompts related to the Internet, such as the following, and have students add their answers to their journals: ▪ How is the Internet like the US Post Office? ▪ Explain the difference between circuit switching and packet switching. Ask students to use what they learned from reading to make a drawing showing how they think an email travels from one place to another.
2	4.3	**Predict and compare** When introducing parallel and distributed computing, present students with a set of processes and several distributed models. Ask students to compare the models and predict which one is the most efficient, least efficient, or equivalent to other models in the set. Then show students how to determine the efficiency of each model to check if their predictions were correct.

Big Idea Planning Notes

Use the space below to plan your approach to the topics in this big idea. Consider what resources and instructional strategies you might want to use.

© 2023 College Board

SKILL

5.A

Explain how computing systems work.

TOPIC 4.1
The Internet

AVAILABLE RESOURCES

- Classroom Resources > **The Computer and the Internet: TCP/IP: How Messages Get Delivered Across the Internet**
- External Resources >
  - **Routing and Deadlock** from CS Unplugged
  - **Network Protocols** from CS Unplugged
  - **Blown to Bits: Appendix**

Required Course Content

ENDURING UNDERSTANDING

CSN-1

Computer systems and networks facilitate the transfer of data.

LEARNING OBJECTIVE

CSN-1.A

Explain how computing devices work together in a network. 5.A

ESSENTIAL KNOWLEDGE

CSN-1.A.1

A *computing device* is a physical artifact that can run a program. Some examples include computers, tablets, servers, routers, and smart sensors.

CSN-1.A.2

A *computing system* is a group of computing devices and programs working together for a common purpose.

CSN-1.A.3

A *computer network* is a group of interconnected computing devices capable of sending or receiving data.

CSN-1.A.4

A computer network is a type of computing system.

CSN-1.A.5

A *path* between two computing devices on a computer network (a sender and a receiver) is a sequence of directly connected computing devices that begins at the sender and ends at the receiver.

CSN-1.A.6

Routing is the process of finding a path from sender to receiver.

continued on next page

© 2023 College Board

LEARNING OBJECTIVE	ESSENTIAL KNOWLEDGE
CSN-1.A Explain how computing devices work together in a network. 5.A	**CSN-1.A.7** The *bandwidth* of a computer network is the maximum amount of data that can be sent in a fixed amount of time. **CSN-1.A.8** Bandwidth is usually measured in bits per second.
CSN-1.B Explain how the Internet works. 5.A	**CSN-1.B.1** The Internet is a computer network consisting of interconnected networks that use standardized, open (nonproprietary) communication protocols. **CSN-1.B.2** Access to the Internet depends on the ability to connect a computing device to an Internet-connected device. **CSN-1.B.3** A *protocol* is an agreed-upon set of rules that specify the behavior of a system. **CSN-1.B.4** The protocols used in the Internet are *open*, which allows users to easily connect additional computing devices to the Internet. **CSN-1.B.5** Routing on the Internet is usually dynamic; it is not specified in advance. **CSN-1.B.6** The *scalability* of a system is the capacity for the system to change in size and scale to meet new demands. **CSN-1.B.7** The Internet was designed to be scalable.

continued on next page

© 2023 College Board

Computer Systems and Networks

LEARNING OBJECTIVE	ESSENTIAL KNOWLEDGE
CSN-1.C Explain how data are sent through the Internet via packets. **5.A**	**CSN-1.C.1** Information is passed through the Internet as a *data stream*. Data streams contain chunks of data, which are encapsulated in *packets*.
	CSN-1.C.2 Packets contain a chunk of data and metadata used for routing the packet between the origin and the destination on the Internet, as well as for data reassembly.
	CSN-1.C.3 Packets may arrive at the destination in order, out of order, or not at all.
	CSN-1.C.4 IP, TCP, and UDP are common protocols used on the Internet.
CSN-1.D Describe the differences between the Internet and the World Wide Web. **5.A**	**CSN-1.D.1** The World Wide Web is a system of linked pages, programs, and files.
	CSN-1.D.2 HTTP is a protocol used by the World Wide Web.
	CSN-1.D.3 The World Wide Web uses the Internet.

© 2023 College Board

TOPIC 4.2
Fault Tolerance

SKILLS

1.D
Evaluate solution options.

5.A
Explain how computing systems work.

Required Course Content

ENDURING UNDERSTANDING

CSN-1
Computer systems and networks facilitate the transfer of data.

LEARNING OBJECTIVE

CSN-1.E
For fault-tolerant systems, like the Internet:

- **a.** Describe the benefits of fault tolerance. 1.D
- **b.** Explain how a given system is fault-tolerant. 5.A
- **c.** Identify vulnerabilities to failure in a system. 1.D

ESSENTIAL KNOWLEDGE

CSN-1.E.1
The Internet has been engineered to be fault-tolerant, with abstractions for routing and transmitting data.

CSN-1.E.2
Redundancy is the inclusion of extra components that can be used to mitigate failure of a system if other components fail.

CSN-1.E.3
One way to accomplish network redundancy is by having more than one path between any two connected devices.

CSN-1.E.4
If a particular device or connection on the Internet fails, subsequent data will be sent via a different route, if possible.

CSN-1.E.5
When a system can support failures and still continue to function, it is called *fault-tolerant*. This is important because elements of complex systems fail at unexpected times, often in groups, and fault tolerance allows users to continue to use the network.

CSN-1.E.6
Redundancy within a system often requires additional resources but can provide the benefit of fault tolerance.

continued on next page

© 2023 College Board

Computer Systems and Networks

LEARNING OBJECTIVE

CSN-1.E

For fault-tolerant systems, like the Internet:

a. Describe the benefits of fault tolerance. 1.D

b. Explain how a given system is fault-tolerant. 5.A

c. Identify vulnerabilities to failure in a system. 1.D

ESSENTIAL KNOWLEDGE

CSN-1.E.7

The redundancy of routing options between two points increases the reliability of the Internet and helps it scale to more devices and more people.

© 2023 College Board

TOPIC 4.3
Parallel and Distributed Computing

SKILL

1.D

Evaluate solution options.

Required Course Content

ENDURING UNDERSTANDING

CSN-2

Parallel and distributed computing leverage multiple computers to more quickly solve complex problems or process large data sets.

LEARNING OBJECTIVE

CSN-2.A

For sequential, parallel, and distributed computing:

a. Compare problem solutions. 1.D

b. Determine the efficiency of solutions. 1.D

ESSENTIAL KNOWLEDGE

CSN-2.A.1

Sequential computing is a computational model in which operations are performed in order one at a time.

CSN-2.A.2

Parallel computing is a computational model where the program is broken into multiple smaller sequential computing operations, some of which are performed simultaneously.

CSN-2.A.3

Distributed computing is a computational model in which multiple devices are used to run a program.

CSN-2.A.4

Comparing efficiency of solutions can be done by comparing the time it takes them to perform the same task.

CSN-2.A.5

A sequential solution takes as long as the sum of all of its steps.

CSN-2.A.6

A parallel computing solution takes as long as its sequential tasks plus the longest of its parallel tasks.

continued on next page

© 2023 College Board

LEARNING OBJECTIVE	ESSENTIAL KNOWLEDGE
CSN-2.A For sequential, parallel, and distributed computing: **a.** Compare problem solutions. 1.D **b.** Determine the efficiency of solutions. 1.D	**CSN-2.A.7** The "speedup" of a parallel solution is measured in the time it took to complete the task sequentially divided by the time it took to complete the task when done in parallel.
CSN-2.B Describe benefits and challenges of parallel and distributed computing. 1.D	**CSN-2.B.1** Parallel computing consists of a parallel portion and a sequential portion. **CSN-2.B.2** Solutions that use parallel computing can scale more effectively than solutions that use sequential computing. **CSN-2.B.3** Distributed computing allows problems to be solved that could not be solved on a single computer because of either the processing time or storage needs involved. **CSN-2.B.4** Distributed computing allows much larger problems to be solved quicker than they could be solved using a single computer. **CSN-2.B.5** When increasing the use of parallel computing in a solution, the efficiency of the solution is still limited by the sequential portion. This means that at some point, adding parallel portions will no longer meaningfully increase efficiency.

© 2023 College Board

AP COMPUTER SCIENCE PRINCIPLES

BIG IDEA 5

Impact of Computing

21–26%
AP EXAM WEIGHTING

© 2023 College Board

Remember to go to **AP Classroom** to assign students the online Topic Questions for this big idea.

Whether assigned as homework or completed in class, the Topic Questions can be used to spot-check student understanding and help identify content and skills to emphasize in lessons.

Topic Questions

Multiple-choice: ~20 questions

© 2023 College Board

BIG IDEA 5

21–26% AP EXAM WEIGHTING

Impact of Computing

ESSENTIAL QUESTIONS

IOC-1

- What app or computer software do you use most often and would have a hard time going without? How does this software solve a problem for you or benefit you?
- Are innovators responsible for the harmful effects of their computing innovations, even if those effects were unintentional? Why or why not?

IOC-2

- What data are generated by smart phones, and what are they being used for?

Developing Understanding

The creation of computer programs can have extensive impacts, some unintended, on societies, economies, and cultures. In this big idea, students explore these effects, the legal and ethical concerns that come with programs, and the responsibilities of programmers. When using computing innovations and transmitting information via the Internet, students should be aware of the risk of sharing personal identifiable information about themselves, such as their age or address, and actively take steps to keep this information safe. This big idea can be integrated throughout the course and works well with the Creative Development, Data, and Computing Systems and Networks big ideas.

Building Computational Thinking Practices

1.C 5.C 5.D 5.E

Computing innovations and programs are often developed in teams. A good, collaborative team starts with a group that is made up of people from different backgrounds, genders, ages, and races so that the perspectives of all potential users are being represented. By creating diverse groups where each person's opinion is considered, we help avoid unintentional bias and potential negative effects, such as contributing to the digital divide, that can creep into innovations.

Investigating the impact of existing computing innovations can help students avoid unintentional negative effects of their own innovations. Consumers should be aware of the impact that a new computing innovation might have before beginning to use it, as well as what data are being gathered and how the product owner intends to use those data. While students may find it relatively easy to describe how the gathering of data would impact them, it is sometimes more difficult for students to understand the impacts that computing innovations and the gathering of personal data might have on people who are different from them, or on society as a whole. As students investigate computing innovations, provide opportunities for students to learn from others' perspectives by allowing time for viewpoints and potential impacts to be shared during a group discussion, like a debate.

Preparing for the AP Exam

Students will be asked to complete three investigations into computing innovations during the school year. Through these investigations, students will look at the data the computing innovation uses to complete its task; any data privacy, security, or storage concerns that might be associated with the innovation; and beneficial and harmful effects the computing innovation might have on society, the economy, or culture.

On the end-of-course exam, students will be presented with a passage about a computing innovation and will be asked a series of questions about data and the effects of the computing innovation. While the computing innovations that need to be investigated are not specified in the curricular requirement, students will benefit from investigating a large range of computing innovations.

© 2023 College Board

BIG IDEA AT A GLANCE

Learning Objective	Topic	Skills	Unit/Module
IOC-1.A, IOC-1.B	**5.1 Beneficial and Harmful Effects**	5.C Describe the impact of a computing innovation.	
IOC-1.C	**5.2 Digital Divide**	5.C Describe the impact of a computing innovation.	
IOC-1.D	**5.3 Computing Bias**	5.E Evaluate the use of computing based on legal and ethical factors.	
IOC-1.E	**5.4 Crowdsourcing**	1.C Explain how collaboration affects the development of a solution.	
IOC-1.F	**5.5 Legal and Ethical Concerns**	5.E Evaluate the use of computing based on legal and ethical factors.	
IOC-2.A, IOC-2.B, IOC-2.C	**5.6 Safe Computing**	5.D Describe the impact of gathering data. 5.E Evaluate the use of computing based on legal and ethical factors.	

Go to **AP Classroom** to assign Topic Questions as you teach the topics in Big Idea 5. Review the results in class to identify and address any student misunderstandings.

© 2023 College Board

SAMPLE INSTRUCTIONAL ACTIVITIES

The sample activities on this page are optional and are offered to provide possible ways to incorporate instructional approaches into the classroom. They were developed in partnership with teachers from the AP community to share ways that they approach teaching some of the topics and skills in this big idea. Please refer to the Instructional Strategies section beginning on p. 132 for more examples of activities and strategies.

Activity	Topic	Sample Activity
1	5.1	**Marking the text** Provide students with an article that highlights both beneficial and harmful effects of a specific computing innovation, and have them mark which effects are beneficial and which are harmful. For each effect the students mark as harmful, have them add notes about whether they think these effects should have been anticipated in advance. For each effect the students mark as beneficial, have the students make notes indicating if they think these benefits were intended or unintended.
2	5.6	**Kinesthetic learning** In small groups, have students create and act out a play or a scene involving privacy and security risks, especially when it comes to personally identifiable information (PII) and the impact of collecting such data. Sample topics might include not recognizing a phishing email, being careless with passwords, downloading a virus accidentally, or not being aware of a search history being kept on a computer. Students could extend their play to include best practices or ways to stay safer when using computing innovations.

Big Idea Planning Notes

Use the space below to plan your approach to the topics in this big idea. Consider what resources and instructional strategies you might want to use.

© 2023 College Board

SKILL

5.C

Describe the impact of a computing innovation.

AVAILABLE RESOURCE

- External Resource > ACM Tech News

TOPIC 5.1
Beneficial and Harmful Effects

Required Course Content

ENDURING UNDERSTANDING

IOC-1

While computing innovations are typically designed to achieve a specific purpose, they may have unintended consequences.

LEARNING OBJECTIVE

IOC-1.A

Explain how an effect of a computing innovation can be both beneficial and harmful.

5.C

ESSENTIAL KNOWLEDGE

IOC-1.A.1

People create computing innovations.

IOC-1.A.2

The way people complete tasks often changes to incorporate new computing innovations.

IOC-1.A.3

Not every effect of a computing innovation is anticipated in advance.

IOC-1.A.4

A single effect can be viewed as both beneficial and harmful by different people, or even by the same person.

IOC-1.A.5

Advances in computing have generated and increased creativity in other fields, such as medicine, engineering, communications, and the arts.

continued on next page

© 2023 College Board

LEARNING OBJECTIVE

IOC-1.B

Explain how a computing innovation can have an impact beyond its intended purpose.

5.C

ESSENTIAL KNOWLEDGE

IOC-1.B.1

Computing innovations can be used in ways that their creators had not originally intended:

- The World Wide Web was originally intended only for rapid and easy exchange of information within the scientific community.
- Targeted advertising is used to help businesses, but it can be misused at both individual and aggregate levels.
- Machine learning and data mining have enabled innovation in medicine, business, and science, but information discovered in this way has also been used to discriminate against groups of individuals.

IOC-1.B.2

Some of the ways computing innovations can be used may have a harmful impact on society, the economy, or culture.

IOC-1.B.3

Responsible programmers try to consider the unintended ways their computing innovations can be used and the potential beneficial and harmful effects of these new uses.

IOC-1.B.4

It is not possible for a programmer to consider all the ways a computing innovation can be used.

IOC-1.B.5

Computing innovations have often had unintended beneficial effects by leading to advances in other fields.

IOC-1.B.6

Rapid sharing of a program or running a program with a large number of users can result in significant impacts beyond the intended purpose or control of the programmer.

© 2023 College Board

SKILL

5.C

Describe the impact of a computing innovation.

AVAILABLE RESOURCE

- External Resource > ACM Tech News

TOPIC 5.2
Digital Divide

Required Course Content

ENDURING UNDERSTANDING

IOC-1

While computing innovations are typically designed to achieve a specific purpose, they may have unintended consequences.

LEARNING OBJECTIVE

IOC-1.C

Describe issues that contribute to the digital divide.

5.C

ESSENTIAL KNOWLEDGE

IOC-1.C.1

Internet access varies between socioeconomic, geographic, and demographic characteristics, as well as between countries.

IOC-1.C.2

The "digital divide" refers to differing access to computing devices and the Internet, based on socioeconomic, geographic, or demographic characteristics.

IOC-1.C.3

The digital divide can affect both groups and individuals.

IOC-1.C.4

The digital divide raises issues of equity, access, and influence, both globally and locally.

IOC-1.C.5

The digital divide is affected by the actions of individuals, organizations, and governments.

© 2023 College Board

TOPIC 5.3
Computing Bias

SKILL

5.E

Evaluate the use of computing based on legal and ethical factors.

AVAILABLE RESOURCE

- External Resource > ACM Tech News

Required Course Content

ENDURING UNDERSTANDING

IOC-1

While computing innovations are typically designed to achieve a specific purpose, they may have unintended consequences.

LEARNING OBJECTIVE

IOC-1.D

Explain how bias exists in computing innovations. 5.E

ESSENTIAL KNOWLEDGE

IOC-1.D.1

Computing innovations can reflect existing human biases because of biases written into the algorithms or biases in the data used by the innovation.

IOC-1.D.2

Programmers should take action to reduce bias in algorithms used for computing innovations as a way of combating existing human biases.

IOC-1.D.3

Biases can be embedded at all levels of software development.

© 2023 College Board

SKILL

1.C

Explain how collaboration affects the development of a solution.

TOPIC 5.4
Crowdsourcing

Required Course Content

ENDURING UNDERSTANDING

IOC-1

While computing innovations are typically designed to achieve a specific purpose, they may have unintended consequences.

LEARNING OBJECTIVE

IOC-1.E

Explain how people participate in problem-solving processes at scale. 1.C

ESSENTIAL KNOWLEDGE

IOC-1.E.1

Widespread access to information and public data facilitates the identification of problems, development of solutions, and dissemination of results.

IOC-1.E.2

Science has been affected by using distributed and "citizen science" to solve scientific problems.

IOC-1.E.3

Citizen science is scientific research conducted in whole or part by distributed individuals, many of whom may not be scientists, who contribute relevant data to research using their own computing devices.

IOC-1.E.4

Crowdsourcing is the practice of obtaining input or information from a large number of people via the Internet.

IOC-1.E.5

Human capabilities can be enhanced by collaboration via computing.

IOC-1.E.6

Crowdsourcing offers new models for collaboration, such as connecting businesses or social causes with funding.

© 2023 College Board

TOPIC 5.5
Legal and Ethical Concerns

SKILL

5.E

Evaluate the use of computing based on legal and ethical factors.

AVAILABLE RESOURCES

- Classroom Resources > **Ethical Use of the Computer**
- External Resources >
  - **ACM Tech News**
  - **Blown to Bits: Chapter 7**

Required Course Content

ENDURING UNDERSTANDING

IOC-1

While computing innovations are typically designed to achieve a specific purpose, they may have unintended consequences.

LEARNING OBJECTIVE

IOC-1.F

Explain how the use of computing can raise legal and ethical concerns. 5.E

ESSENTIAL KNOWLEDGE

IOC-1.F.1

Material created on a computer is the intellectual property of the creator or an organization.

IOC-1.F.2

Ease of access and distribution of digitized information raises intellectual property concerns regarding ownership, value, and use.

IOC-1.F.3

Measures should be taken to safeguard intellectual property.

IOC-1.F.4

The use of material created by someone else without permission and presented as one's own is plagiarism and may have legal consequences.

continued on next page

© 2023 College Board

BIG IDEA 5

Impact of Computing

LEARNING OBJECTIVE

IOC-1.F

Explain how the use of computing can raise legal and ethical concerns. 5.E

ESSENTIAL KNOWLEDGE

IOC-1.F.5

Some examples of legal ways to use materials created by someone else include:

- Creative Commons—a public copyright license that enables the free distribution of an otherwise copyrighted work. This is used when the content creator wants to give others the right to share, use, and build upon the work they have created.
- open source—programs that are made freely available and may be redistributed and modified
- open access—online research output free of any and all restrictions on access and free of many restrictions on use, such as copyright or license restrictions

IOC-1.F.6

The use of material created by someone other than you should always be cited.

IOC-1.F.7

Creative Commons, open source, and open access have enabled broad access to digital information.

IOC-1.F.8

As with any technology or medium, using computing to harm individuals or groups of people raises legal and ethical concerns.

IOC-1.F.9

Computing can play a role in social and political issues, which in turn often raises legal and ethical concerns.

IOC-1.F.10

The digital divide raises ethical concerns around computing.

IOC-1.F.11

Computing innovations can raise legal and ethical concerns. Some examples of these include:

- the development of software that allows access to digital media downloads and streaming
- the development of algorithms that include bias
- the existence of computing devices that collect and analyze data by continuously monitoring activities

© 2023 College Board

TOPIC 5.6
Safe Computing

SKILLS

5.D
Describe the impact of gathering data.

5.E
Evaluate the use of computing based on legal and ethical factors.

AVAILABLE RESOURCES
- External Resources >
  - **ACM Tech News**
  - **Public Key Encryption** from CS Unplugged
  - **Blown to Bits: Chapter 2 and 5**

Required Course Content

ENDURING UNDERSTANDING

IOC-2
The use of computing innovations may involve risks to personal safety and identity.

LEARNING OBJECTIVE

IOC-2.A
Describe the risks to privacy from collecting and storing personal data on a computer system. **5.D**

ESSENTIAL KNOWLEDGE

IOC-2.A.1
Personally identifiable information (PII) is information about an individual that identifies, links, relates, or describes them. Examples of PII include:
- Social Security number
- age
- race
- phone number(s)
- medical information
- financial information
- biometric data

IOC-2.A.2
Search engines can record and maintain a history of searches made by users.

IOC-2.A.3
Websites can record and maintain a history of individuals who have viewed their pages.

IOC-2.A.4
Devices, websites, and networks can collect information about a user's location.

IOC-2.A.5
Technology enables the collection, use, and exploitation of information about, by, and for individuals, groups, and institutions.

continued on next page

© 2023 College Board

LEARNING OBJECTIVE

IOC-2.A

Describe the risks to privacy from collecting and storing personal data on a computer system. 5.D

ESSENTIAL KNOWLEDGE

IOC-2.A.6

Search engines can use search history to suggest websites or for targeted marketing.

IOC-2.A.7

Disparate personal data, such as geolocation, cookies, and browsing history, can be aggregated to create knowledge about an individual.

IOC-2.A.8

PII and other information placed online can be used to enhance a user's online experiences.

IOC-2.A.9

PII stored online can be used to simplify making online purchases.

IOC-2.A.10

Commercial and governmental curation of information may be exploited if privacy and other protections are ignored.

IOC-2.A.11

Information placed online can be used in ways that were not intended and that may have a harmful impact. For example, an email message may be forwarded, tweets can be retweeted, and social media posts can be viewed by potential employers.

IOC-2.A.12

PII can be used to stalk or steal the identity of a person or to aid in the planning of other criminal acts.

IOC-2.A.13

Once information is placed online, it is difficult to delete.

IOC-2.A.14

Programs can collect your location and record where you have been, how you got there, and how long you were at a given location.

IOC-2.A.15

Information posted to social media services can be used by others. Combining information posted on social media and other sources can be used to deduce private information about you.

continued on next page

© 2023 College Board

LEARNING OBJECTIVE

IOC-2.B

Explain how computing resources can be protected and can be misused. **5.E**

ESSENTIAL KNOWLEDGE

IOC-2.B.1

Authentication measures protect devices and information from unauthorized access. Examples of authentication measures include strong passwords and multifactor authentication.

IOC-2.B.2

A strong password is something that is easy for a user to remember but would be difficult for someone else to guess based on knowledge of that user.

IOC-2.B.3

Multifactor authentication is a method of computer access control in which a user is only granted access after successfully presenting several separate pieces of evidence to an authentication mechanism, typically in at least two of the following categories: knowledge (something they know), possession (something they have), and inherence (something they are).

IOC-2.B.4

Multifactor authentication requires at least two steps to unlock protected information; each step adds a new layer of security that must be broken to gain unauthorized access.

IOC-2.B.5

Encryption is the process of encoding data to prevent unauthorized access. *Decryption* is the process of decoding the data. Two common encryption approaches are:

- Symmetric key encryption involves one key for both encryption and decryption.
- Public key encryption pairs a public key for encryption and a private key for decryption. The sender does not need the receiver's private key to encrypt a message, but the receiver's private key is required to decrypt the message.

EXCLUSION STATEMENT (EK IOC-2.B.5):
Specific mathematical procedures for encryption and decryption are beyond the scope of this course and the AP Exam.

IOC-2.B.6

Certificate authorities issue digital certificates that validate the ownership of encryption keys used in secure communications and are based on a trust model.

continued on next page

© 2023 College Board

Impact of Computing

LEARNING OBJECTIVE	ESSENTIAL KNOWLEDGE
IOC-2.B Explain how computing resources can be protected and can be misused. 5.E	**IOC-2.B.7** Computer virus and malware scanning software can help protect a computing system against infection. **IOC-2.B.8** A *computer virus* is a malicious program that can copy itself and gain access to a computer in an unauthorized way. Computer viruses often attach themselves to legitimate programs and start running independently on a computer. **IOC-2.B.9** *Malware* is software intended to damage a computing system or to take partial control over its operation. **IOC-2.B.10** All real-world systems have errors or design flaws that can be exploited to compromise them. Regular software updates help fix errors that could compromise a computing system. **IOC-2.B.11** Users can control the permissions programs have for collecting user information. Users should review the permission settings of programs to protect their privacy.
IOC-2.C Explain how unauthorized access to computing resources is gained. 5.E	**IOC-2.C.1** *Phishing* is a technique that attempts to trick a user into providing personal information. That personal information can then be used to access sensitive online resources, such as bank accounts and emails. **IOC-2.C.2** *Keylogging* is the use of a program to record every keystroke made by a computer user in order to gain fraudulent access to passwords and other confidential information. **IOC-2.C.3** Data sent over public networks can be intercepted, analyzed, and modified. One way that this can happen is through a rogue access point. **IOC-2.C.4** A *rogue access point* is a wireless access point that gives unauthorized access to secure networks.

continued on next page

© 2023 College Board

LEARNING OBJECTIVE

IOC-2.C

Explain how unauthorized access to computing resources is gained. 5.E

ESSENTIAL KNOWLEDGE

IOC-2.C.5

A malicious link can be disguised on a web page or in an email message.

IOC-2.C.6

Unsolicited emails, attachments, links, and forms in emails can be used to compromise the security of a computing system. These can come from unknown senders or from known senders whose security has been compromised.

IOC-2.C.7

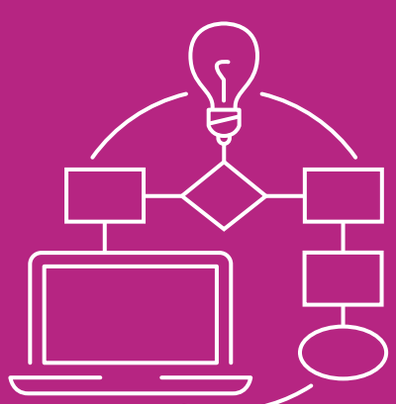
Untrustworthy (often free) downloads from freeware or shareware sites can contain malware.

© 2023 College Board

THIS PAGE IS INTENTIONALLY LEFT BLANK.

AP COMPUTER SCIENCE PRINCIPLES

Instructional Approaches



Selecting and Using Course Materials

AP Endorsed Curriculum and Professional Development Providers

College Board has endorsed several curriculum and professional development providers for AP Computer Science Principles. AP endorsed providers offer teachers a complete curriculum, including daily lesson plans and activities, that is completely aligned to the AP Computer Science Principles course framework, as well as an aligned syllabus and professional development. Each AP endorsed provider is unique in the way it structures the curriculum, chooses a programming language, and provides support for teachers. When selecting a provider, teachers will want to consider how comfortable they are with the content, the computer science background of their students, and any potential costs for the curriculum and professional development. Once an endorsed provider is selected, teachers can adopt an endorsed provider syllabus by selecting it on the **AP Course Audit** website. A complete list of current AP CSP endorsed providers can be found the on the **Classroom Resources** page on AP Central.

Course Pacing

Teachers can choose to create their own curriculum and syllabus based on the AP Computer Science Principles course framework. Because AP CSP is not presented in units, teachers who choose to create their own curriculum and syllabus will need to develop unit plans to integrate and scaffold the big ideas and skills in this course. Teachers will also need to select a programming language that is appropriate for the course and their students. The Curriculum Alignment section of the course and exam description contains additional guidance on how to sequence your course along with a Unit at a Glance page that can be replicated to plan which topics you intend to cover in that unit. When determining the number of class periods to spend on each topic, be sure to include time for students to complete the corresponding formative Topic Questions in **AP Classroom**. Since students will need to spend 9 hours in class on the Create performance task, teachers will want to be sure to include this time in their planning.

If you choose an AP endorsed provider, check to see if they have developed their own Unit at a Glance chart or something similar that can be printed and added to the Curriculum Alignment section. This document should align the provider curriculum to the AP course and exam description and formative Topic Questions in AP Classroom.

Choosing a Programming Language

AP Computer Science Principles does not require the use of a specific programming language. Because a goal of this course is to broaden participation, teachers can center their course around computing concepts in the course framework that support the creation of exciting and relevant computational artifacts. For this reason, teachers are encouraged to select the programming language that is most appropriate for their classroom and that will provide students opportunities to successfully engage with the course content. The programming language selected should contain functionality that is specified in the course framework and performance tasks. Appropriate programming languages for this course are ones that allow students to evaluate expressions, develop procedures, and use variables, lists, conditionals, and loops.

© 2023 College Board

The following is a noncomprehensive list of programming languages or development environments that can be considered for use in this course:

Language/Product	Description
Alice	This block-based programming language includes a 3-D modeling environment that allows students to create and animate 3-D worlds. This environment lends itself well to creating stories and games.
App Inventor	This open-source web application is block-based and allows students to create their own applications on mobile devices.
App Lab	This is a programming environment for creating web applications with JavaScript. It allows students to develop programs and toggle back and forth between block-based and text-based programming modes.
EarSketch	This text-based browser application allows students to create their own music using either JavaScript or Python.
Greenfoot	This text-based Java IDE is designed for use in education to create 2-D graphic applications, such as simulations and interactive games.
Java	This text-based programming language allows students to create and solve problems that vary widely in difficulty. There are several IDEs that can be used to write programs in Java.
JavaScript	This text-based programming language is commonly used to create interactive effects within web browsers.
LEGO Mindstorms EV3	This product integrates block-based programming with LEGO bricks and sensors to create and program robots. The instructions are assembled by linking together function blocks.
Microsoft MakeCode	This development environment provides both block and text editors for students at different levels. Microsoft MakeCode is a free, open-source, web-based environment with open educational resources for teachers.
Processing	This text-based programming language was initially created to serve as a software sketchbook, and it can be used to teach programming in a visual context.
Python	This text-based programming language has the benefit of readability, which might be helpful to new programmers.
Quorum	This universally designed and evidence-based programming language allows students, including those with disabilities, to create programs for gaming, robotics, sound processing, networking, and more.
Scratch	This block-based programming language allows students to build scripts to run animations. This product can be downloaded and installed on a computer or run in a browser.
Snap*!*	Snap*!* combines the power of text-based languages such as Python or JavaScript with the visual simplicity of block-based Scratch. It can grow with your students, because new blocks can be written in Snap*!* itself or in JavaScript.
Swift	This powerful and intuitive programming language can be used for macOS, iOS, watchOS, tvOS and beyond. Students gain practical experience with the tools and techniques to build basic iOS apps with Swift and Xcode, an IDE at the center of the Apple development experience.
TI-Basic	This is an introductory text-based programming language for TI graphing calculators. Programs with the TI-Innovator Hub can read inputs from sensors and send output to control speakers, LEDs, motors, a robotic vehicle, and more.

NOTE: *While teachers may choose to do some programming instruction using HTML, it should be noted that HTML is not an acceptable programming language to use when completing the Create performance task.*

© 2023 College Board

This course provides students with an AP Computer Science Principles Exam Reference Sheet, as seen in the Appendix, to use when taking the AP Exam. The Exam Reference Sheet is not meant to be a substitute for choosing a programming language that is recognized by the field of computer science or being used in postsecondary institutions. However, because there is no designated programming language for the course, the Exam Reference Sheet was developed to provide a common programming language to be used when assessing students' skills and knowledge of the programming constructs described in the course framework. It is meant to establish a common way to communicate programming concepts for the purpose of the exam. Teachers are encouraged to integrate the use of the Exam Reference Sheet throughout the school year.

© 2023 College Board

Instructional Strategies

The AP Computer Science Principles course framework details the concepts and skills students must master to be successful on the AP Exam. To address those concepts and skills effectively, it helps to incorporate a variety of instructional approaches and best practices into daily lessons and activities. The following table presents strategies that can help students develop mastery of the skills by engaging them in learning activities that apply their understanding of course concepts. The instructional strategies have been categorized based on whether they are used to teach programming and problem-solving, are suitable for cooperative learning, or are helpful for students to make connections between material being presented and prior knowledge of topics covered.

Programming and Problem-Solving

Strategy	Definition	Purpose	Example
Code tracing	Students go step-by-step through program code by hand to determine how a piece of program code operates.	Prepares students to determine the output of program code, as well as detect and correct errors in code, when a computer is not available.	Have students create a multicolumn chart with a column to record the value of each variable after each iteration of a program. Then, provide students with several code segments featuring iteration, and have them practice code tracing by recording any values that are changed, printed, or returned.
Create a plan	Students analyze the tasks in a problem and create a process for completing the tasks by finding the information needed, interpreting data, choosing how to solve a problem, communicating results, and verifying accuracy.	Assists in breaking programming tasks into smaller parts and identifying the steps needed to complete the entire task. One example of this is the different phases of an iterative development process and how they work together in the completion of a program.	Have students brainstorm several problems in their own lives or in their communities. Then, have them select one of the problems and conduct research to gather more of an understanding of that problem and identify at least five facts that would help find a solution to the problem. Next, have students brainstorm at least three possible app ideas that might help solve the identified problem and describe how the apps might function. Finally, have students select one of the apps and describe how would they determine if it was successful or not.

continued on next page

© 2023 College Board

Programming and Problem-Solving (cont'd)

Strategy	Definition	Purpose	Example
Error analysis	Students analyze an existing solution to determine whether (or where) errors have occurred.	Allows students to troubleshoot common errors that may arise when they use similar methods and focus on correcting these errors.	When students begin to write procedures, have them analyze the method output and troubleshoot any errors that might lead to incorrect output or runtime errors. By becoming more familiar with the types of errors that may happen, students should be better able to spot them in unfamiliar program code.
Identify a subtask	Students break a problem into smaller pieces whose outcomes lead to a solution.	Helps organize the pieces of a complex problem and reach a complete solution.	When students use abstraction to manage complexity of their program during program development, have them look at the subtasks that might exist in a solution and create procedures with parameters to generalize that functionality to work with a greater set of values.
Look for a pattern	Students observe trends by looking at expected output or results based on input and specifications.	Helps identify patterns that can be used to design program code, generalize program behavior, identify errors in existing program code, or draw conclusions from data.	Provide students with multiple representations of the same set of data, such as bar graphs, pie charts, and word clouds. Have students analyze the data to detect patterns. They can then use the patterns they detected to draw conclusions.
Marking the text	Students highlight, underline, and/or annotate text to focus on key information and help understand the text or solve the problem.	Helps students identify important information in the text of a program specification or resource about a computing innovation and make notes about the interpretation of tasks required to implement the program specification or how the computing innovation works.	When students are reading a passage about a computing innovation, have them focus on marking the text by circling any information that is connected to data and underlining any information relevant to the impact of the innovation.

continued on next page

© 2023 College Board

Programming and Problem-Solving (cont'd)

Strategy	Definition	Purpose	Example
Pair programming	Two programmers work together as a pair. One (the driver) writes program code, while the other (the observer, pointer, or navigator) reviews each line of program code as it is typed in.	Reinforces the need for students to explain their process in a way that another student can understand. It also provides built-in support as students practice collaborating while learning new material.	Provide students with program code that uses iteration to draw a square. Then, have them work in pairs to modify the program code to draw other shapes, like a triangle, pentagon, hexagon, octagon, or circle. Have students switch who drives the program after each shape. Ask them to identify patterns between these shapes and the commands used.
Predict and compare	Students make conjectures about what results will develop in an activity, confirming or modifying the conjectures based on outcomes.	Stimulates thinking by making, checking, and correcting predictions of program output based on evidence from the outcome.	Provide students working in groups with partially written program code and options for completing the code that would move a robot through a maze to its destination. Have students predict which code would allow the robot to successfully reach its goal. After making their predictions, have students act out the code as a human robot on a grid made of masking tape on the floor. Finally, ask students to compare their predictions to the actual outcomes and discuss where their predictions went wrong.
Simplify the problem	Students use simpler numbers or statements to solve a problem.	Provides insight into a more abstract problem by making it concrete and allowing students to more easily recognize a general process to obtain a solution.	When developing an algorithm to analyze data, consider a small data set as a first example to confirm that your process will yield the proper results.
Think aloud	Students talk through a difficult problem by describing what the text or code means.	Engages students with a problem in a new way that puts them in the role of thinking of potential solutions aloud. This trains students to consider solutions for themselves prior to seeking assistance from a teacher or peer.	When asking students to describe the purpose of a code segment, provide them with a rubber duck or other inanimate object to which they can describe the code segment aloud.

continued on next page

© 2023 College Board

Programming and Problem-Solving (cont'd)

Strategy	Definition	Purpose	Example
Work backward	Students trace a possible answer back through the solution process to the starting point.	Gives students who might not know how to begin to solve a problem a different way to think about it by starting at the ideal solution and breaking it down.	When working with large data sets, have students start by thinking about possible conclusions that can be drawn from the available data. From this list, ask them to identify a few questions that they will want to answer with the data. Finally, have them determine the process they would use to analyze their data to come to the proposed conclusion to test their hypothesis.

Cooperative Learning

Strategy	Definition	Purpose	Example
Discussion group	Students engage in an interactive, small-group discussion.	Allows students to gain new understanding of or insight into a problem or solution by listening to multiple perspectives.	Find a current-events article related to data privacy or security issues and pose a purposeful question to students about an issue in the article. Have students take part in a debate within small groups. Then, ask each group to take a stance on the issue and provide arguments for and against that stance. Finally, as a whole-class activity, have students break into two larger groups of opposing viewpoints.
Jigsaw	Each student in a group reads a different text or passage, taking on the role of "expert" on what was read. Students share the information from that reading with students from other groups and then return to their original groups to share their new knowledge.	Allows students to summarize and present information to others in a way that facilitates understanding of a topic without having each student read the text in its entirety; by teaching others, they become experts.	Have different students read about computing innovations in different fields, such as medicine, engineering, communications, and the arts. Have these students share with students who read about a different field.

continued on next page

© 2023 College Board

Cooperative Learning (cont'd)

Strategy	Definition	Purpose	Example
Kinesthetic learning	Students' body movements are used to create knowledge or understanding of a new concept. A kinesthetic–tactile learning style requires students to manipulate or touch materials to learn.	Allows students to gain new understanding of or insight into a problem or solution by acting out the steps taken when executing a code segment.	When teaching Boolean operators, have all students stand. Present them with a personal attribute or description they can answer yes or no to, such as "wearing red" or "is in the 11th grade." If the statement applies to the student, they remain standing. If the statement does not apply to the student, they sit down. Repeat the process with several statements. Then, challenge the students to combine two of the statements using NOT, AND, or OR. Determine if the statements are logically equivalent based on who remains standing.
Sharing and responding	Students communicate with another person or a small group of peers who respond to a proposed problem or solution.	Gives students the opportunity to discuss their work with peers, make suggestions for improvements to the work of others, and/or receive appropriate and relevant feedback on their own work.	Have students write an algorithm with the stipulation that it cannot include pictures or code. Next, have them share their work with a partner and receive feedback on which parts were unclear or needed improvement.
Think-pair-share	Students think through a problem alone, pair with a partner to share ideas, and then share results with the class.	Enables the development of initial ideas that are then tested with a partner in preparation for revising and then sharing the ideas with a larger group.	Provide students with examples of real-life abstractions, such as how a car key hides the complexity of what occurs under the hood. Have students consider what abstractions they encounter in everyday life. Then, working in pairs, ask students to share examples of everyday abstractions with their partners and compile a larger list before sharing with the whole class.

continued on next page

© 2023 College Board

Cooperative Learning (cont'd)

Strategy	Definition	Purpose	Example
Unplugged activities	Students use engaging games and puzzles that use manipulatives and kinesthetic learning activities.	Provides students with a different way of engaging with the material, away from the computer, to further understanding.	When teaching searching, choose 15 students to line up at the front of the classroom. Provide each student with a card with a number on it (in random order) that they keep hidden. Select a 16th student to be the guesser. Give the guesser a container with five pieces of candy in it. Their job is to find a number provided by the teacher among the group of 15 students. They can use their candy as currency to "pay" one of the students at the front of the classroom to look at their card. If they find the correct number before using all their candies, they get to keep the rest. Then, repeat this process, but now with the students sorted in ascending order to encourage strategy development for finding the correct number.
Using manipulatives	Students use objects to examine relationships in the information given.	Provides a tactile or visual representation of data or processes that support comprehension of information in a problem or concept.	When illustrating public key encryption, provide each small group of students with a box that requires the use of a combination code to unlock. Tell students that you used a public key to lock a message in the box for them to read. Each student in the group should be provided with their own private key code. Only one of the students should have the correct code to unlock the box. Debrief with students about the difference between public key and private key.

© 2023 College Board

Making Connections

Strategy	Definition	Purpose	Example
Activating prior knowledge	The teacher provides students with an opportunity to recall what they already know about a concept and make connections to current studies.	Prepares students to establish content connections.	Have students practice using mathematical operations by writing pseudocode to carry out common computational operations. For example, they can determine the area of a triangle when given values for the base and the height, or determine the flight time between two cities when given the distance between them and the average speed of the airplane.
Diagramming	Students use a visual representation to organize information.	Builds comprehension and facilitates discussion by representing information in visual form.	When students are planning their code, have them diagram or create flowcharts of their algorithms on chart paper. Students can work collaboratively and can use the diagram as a space to make mistakes and edit.
Journaling	Students keep a journal or log of their program progress, including any difficulties or opportunities that might arise, as well as potential next steps.	Provides students with an opportunity to reflect on their progress and to keep track of what they have accomplished and what work is remaining.	While completing a long programming project, have students write a journal entry each day about what they have accomplished, what successes they've had, what difficulties they encountered, and any ideas for changes that came up during that class that they would like to work on in the next class. This activity is especially helpful when completing the Create performance task.
Paraphrase	Students restate, in their own words, essential information expressed in a text.	Assists with comprehension, recall of information, and problem-solving.	Have students read a set of instructions for a program. Allow a volunteer student to explain the program requirements to the other students, ensuring that the student does not leave out any key points.

continued on next page

© 2023 College Board

Making Connections (cont'd)

Strategy	Definition	Purpose	Example
Quickwrite	Students write for a short, specific amount of time about a designated topic related to a given prompt.	Generates multiple ideas in a quick fashion.	Have students investigate a computing innovation that they are familiar with. Ask them to take five minutes to write about all the possible harmful effects of the computing innovation. Then ask them to take five minutes to write about all the possible beneficial effects of the computing innovation.
Vocabulary organizer	Students use a graphic organizer with a designated format to maintain an ongoing record of vocabulary words with definitions, pictures, notations, and connections.	Provides reinforcement of learned words and a personal, ever-present tool for building word knowledge and awareness.	At the end of each unit or module, review the vocabulary words and definitions that students should know. To build an organizer, have students come up with their own examples or pictures that will help them remember the terms later in the course. There are several apps and websites that help students create flashcards and quizzes to help organize the vocabulary they learn throughout the year.

© 2023 College Board

Developing Computational Thinking Practices

Throughout the AP Computer Science Principles course, students will develop computational thinking practices that are fundamental to the discipline of computer science. Because these computational thinking practices represent the complex skills demonstrated by adept computer scientists, students will benefit from multiple opportunities to develop these skills in a scaffolded manner. All questions on the AP Exam and rubric rows in the scoring guidelines for the Create performance task will be associated with one of the skills in the tables that follow, so providing students with practice applying these skills will help prepare them for the exam and set them up for success. The sample exam questions at the end of this course and exam description show how the questions relate to specific computational thinking practices.

The following tables present each skill, key tasks for that skill that relate to how that skill is assessed, questions to aid in determining students' level of understanding, and instructional notes that show ways to address the skill or misconceptions students may have. Skills in Practice 6 are not formally assessed, as they are more observational in nature. Students should demonstrate these skills in class throughout the school year.

Practice 1: *Computational Solution Design*

Skills	Tasks/Questions	Instructional Notes	Sample Activities and Strategies
1.A: Investigate the situation, context, or task.	Analyze computing innovations and problems to: ▪ Identify the purpose of the computing innovation. ▪ Determine whether a problem can be solved with a computing innovation. ▪ Determine what programming constructs might be useful, as well as the necessary requirements for solving the problem. **Key questions for students:** 1. Is the purpose of the computing innovation to solve a problem, provide entertainment for the user, or be creative for the developer? 2. If we are solving a problem, is a computer the right tool to use, or can this task be accomplished easily by hand? 3. What additional information is needed before creating this computing innovation? Do these data need to be represented in a particular way?	Students should recognize that not all problems can be solved with a computer. Additionally, even when computers can solve a problem, they might not be the best tool to use. Determining if simulations, searches, or other programming constructs might be useful requires careful consideration of requirements prior to implementation.	**Jigsaw** Students are divided into groups, and each group is assigned a simulation to become an expert on through testing it out, reading about it, watching videos on it, or analyzing other stimulus materials (e.g., simulations of traffic, stock markets, weather). Student experts will then share their findings in jigsaw groups with students who have explored different simulations. After sharing, ask students to discuss why these simulations are useful, their purpose, and their real-world implications.

continued on next page

© 2023 College Board

Practice 1: *Computational Solution Design (cont'd)*

Skills	Tasks/Questions	Instructional Notes	Sample Activities and Strategies
1.B: Determine and design an appropriate method or approach to achieve the purpose.	Choose a topic or problem to solve and then design a program. This includes designing any user interface that might be required. **Key questions for students:** 1. What are the subparts of the program that should be grouped together? 2. How will you plan out your program? Will you draw out the interface? Will you use pseudocode or flowchart diagrams? 3. How will you know if your program is working?	Students need practice with programming challenges of increasing complexity that relate to ideation and design, with feedback, prior to the official administration of the Create performance task. Students should spend time planning up front before starting their Create performance task. This saves time when implementing the program, as logic errors can be identified and solved much earlier in the process, preventing additional rework.	**Work backward** Before beginning to design a program, ask students to make a table of the different ways they expect the program to behave or of the output they expect to see when it is run. Then, have them add in sample input that should yield these results. Once students have developed their programs, use what they wrote in the table to help confirm if their program is working properly.
1.C: Explain how collaboration affects the development of a solution.	When presented with a scenario, explain effective collaborative strategies. **Key questions for students:** 1. Why is collaboration important? 2. What are the qualities of a good collaborative team? 3. How will you know if your group has been successfully collaborating?	Spend time having students reflect on their collaboration with peers, including what went well and what could be improved in the future. Focus on where the collaboration turned into group work where students each worked individually on part of the project, rather than truly working together and sharing ideas.	**Predict and compare** Have students predict how long it will take to complete various tasks in a scavenger hunt as an individual, as a small group, and as a large group. Next, carry out the tasks with the groups of various sizes. Afterwards, ask them to compare the results of the predictions with the actual outcomes. As you debrief with students, relate this activity to crowdsourcing and the advantages and disadvantages of completing tasks in groups of various sizes.

continued on next page

© 2023 College Board

Practice 1: *Computational Solution Design (cont'd)*

Skills	Tasks/Questions	Instructional Notes	Sample Activities and Strategies
1.D: Evaluate solution options.	When presented with program code or when designing a program: ▪ Compare and contrast different solutions to determine if they yield the same result and which is the best one to use to solve a problem. ▪ Compare the runtimes of different solutions. ▪ Describe the benefits and vulnerabilities of one solution over another. **Key questions for students:** 1. Will different solutions yield the same or similar result? 2. Will one solution be faster than another? 3. What are the benefits of this solution? What might be the limitations?	Exposure to multiple solutions to a single problem allows students to see new ways to approach problems that otherwise might not have been considered. Parallel and distributed computing can seem too abstract for students at first. Providing real-world examples of how distributing solutions has an impact on the amount of time it takes to complete a task can help make this concept more concrete.	**Think-pair-share** Provide students with a diagram of connected computing devices. Some of the devices should be connected to the rest with two or more connections. After having some time to consider the diagram on their own, have students work in pairs to determine the most vulnerable connections in the system and where connections could be added to make the system more fault-tolerant. Allow time for students to share with the group where they feel connections should be added.

Practice 2: *Algorithms and Program Development*

Skills	Tasks/Questions	Instructional Notes	Sample Activities and Strategies
2.A: Represent algorithmic processes without using a programming language.	Express or interpret the meaning of an algorithm that is expressed using a diagram or pseudocode. **Key questions for students:** 1. Is a diagram or pseudocode better to represent this process? 2. How do you know if this representation, when implemented, would lead to the desired result?	Students need exposure to multiple representations—in pseudocode, diagrams, or blocks—of a given algorithm.	**Sharing and Responding** Provide students a set of five to 10 index cards with different numbers on them, placing the cards faceup. In small groups, ask students to write algorithms in pseudocode to determine the highest and lowest cards in the set. Ask students to consider using an algorithm in pseudocode for sorting all the cards in order from lowest to highest.

continued on next page

© 2023 College Board

Practice 2: *Algorithms and Program Development (cont'd)*

Skills	Tasks/Questions	Instructional Notes	Sample Activities and Strategies
2.B: Implement and apply an algorithm.	Write program code that includes sequencing, selection, iteration, and procedures or lists. Given a scenario or partial program code expressed using the Exam Reference Sheet, select the program code that would satisfy the scenario or complete the program code. Apply algorithms by hand, such as to convert from binary to decimal or to search for a value in a set. **Key questions for students:** 1. What programming language is best for this project? 2. Would sorting or filtering the data make them easier to understand? 3. Are there existing algorithms that could be combined to have the desired effect?	Prior to the official administration of the Create performance task, students need practice implementing programs that first include sequencing, then incorporate selection and iteration, and finally use procedures and lists. It is important that students have practice writing program code from scratch, as well as reading and adding to program code that someone else has written.	**Pair programming** Have students use pair programming to write two procedures, switching the driver and navigator for each procedure. The first procedure they write should convert a decimal value to its binary equivalent. The second procedure should convert a binary number to its decimal equivalent.

Practice 3: *Abstraction in Program Development*

Skills	Tasks/Questions	Instructional Notes	Sample Activities and Strategies
3.A: Generalize data sources through variables.	Identify the input, output, and other necessary data in a program, and represent these data using variables. **Key questions for students:** 1. Are there constants being used when a variable would make the code more flexible? 2. How could you use a list to represent something new in your program?	Although there are similarities to the way variables are used in math and computer science, there are also some key differences. In algebra, variables are often used to represent an unknown value or set of values. In computer science, variables hold specific assigned values that can change over time. Students will need multiple opportunities to create and analyze variables to help make this distinction.	**Using manipulatives** Provide students with a set of short problems that could be solved using a computer along with a set of potential data inputs. Have students match the problem with the data that might be necessary to solve the problem.

continued on next page

© 2023 College Board

Practice 3: *Abstraction in Program Development (cont'd)*

Skills	Tasks/Questions	Instructional Notes	Sample Activities and Strategies
3.B: Use abstraction to manage complexity in a program.	Develop and use procedural abstractions by writing procedures that often contain parameters to allow for more general use. Develop data abstractions by using lists and writing program code that is general enough to work even if the list needs to be resized at a later point in time. **Key questions for students:** 1. Are there subtasks that could be pulled out of the solution for reuse? 2. Are you using the same statements in your program repeatedly, just with different values? Would a procedure with parameters work to streamline your code?	Students should understand that simply using a procedure or list does not mean the program code has been generalized. The goal is for the program code to be flexible enough to use in a variety of situations or to still function correctly as the program requirements evolve and change.	**Code tracing** Provide students with two code segments that output a nursery rhyme, such as "Row, Row, Row Your Boat." The first code segment should only contain output statements for each line of the song. The second code segment should call separate procedures for the verses and chorus. As students trace each code segment, they should record both the output and the procedure that is generating the output. Have students compare their code tracing and explain how the code segment that used procedures helped manage the complexity of the program code.
3.C: Explain how abstraction manages complexity.	After writing a program, explain how the use of data abstractions make program code less complex. Given data, explain how those data can be represented in the computer differently by using the binary number system. **Key questions for students:** 1. How does using an abstraction make the program code easier to write, understand, or modify? 2. How could your program code be written more generally so that a change to the size of a list wouldn't necessitate a change to the rest of your program code? 3. How does the use of parameters help make your procedure more usable in new situations?	Using large data sets, in which the solutions are difficult to determine by hand, can help students understand the necessity for and power of data abstractions.	**Journaling** After students write a program, have them identify any abstraction that was used and write a journal entry explaining how that abstraction managed complexity in their program and how they would've needed to write the program differently if they didn't use the abstraction. If the student did not use any abstraction, they should explain where in their program abstraction could've been used to manage complexity.

© 2023 College Board

Practice 4: *Code Analysis*

Skills	Tasks/Questions	Instructional Notes	Sample Activities and Strategies
4.A: Explain how a code segment or program functions.	Given provided program code: ▪ Describe the general behavior of the program code. For example, it keeps the score for the game. ▪ Explain, in detailed steps, how the program code accomplishes this task. For example, the program uses a conditional statement to determine if player A's answer is correct, and if so, it adds 1 to player A's score. **Key questions for students:** 1. What does this program code segment do? 2. What are the individual steps that are being taken in this program code segment?	Students should understand the distinction between "describing what" a code segment does and "explaining how" it accomplishes this task. Often, students answer with a description or an explanation, but not both.	**Paraphrase** Provide students with different code segments. Have them explain what each line or expression does individually. Then have students describe in general what the code segment accomplishes.
4.B: Determine the result of code segments.	Given program code, determine the value stored in a variable or the result produced, which could be graphical or involve robots on a grid. **Key questions for students:** 1. What is the output of this program or code segment? 2. What are all the values that have been stored in this variable throughout the running of the program? 3. What is the final value(s) stored in the variable or list?	Program code on the AP Exam is represented as either text-based or block-based programming code, which can be found on the Exam Reference Sheet. When evaluating the result of a program code segment, execution order matters. Often, questions about higher-level programming concepts involve the use of introductory programming concepts.	**Think-pair-share** Provide students with a code segment that uses iteration and selection to output all the even numbers from 1 to 100. In pairs, have students modify the code segment to output a different number sequence. Once they are done, pairs can create one slide with their program code to be posted for the rest of the class to determine each output.

continued on next page

© 2023 College Board

Practice 4: *Code Analysis (cont'd)*

Skills	Tasks/Questions	Instructional Notes	Sample Activities and Strategies
4.C: Identify and correct errors in algorithms and programs, including error discovery through testing.	Given program code, identify logic errors either through analyzing the program code itself or hand tracing using a set of test data. Determine sample input data for testing a program and describe the expected result each input would produce. **Key questions for students:** 1. When encountering a runtime error, is there anything wrong with the syntax? Has the compiler given you information related to the type of error or the location of the error? 2. For your sample input, does your program produce the expected output? 3. Have you considered all boundary cases to test your algorithm? For example, if you have the condition x < 5, have you tested values just less than 5, equal to 5, and just greater than 5?	Students will encounter many logic and syntax errors while writing programs. Having a collaborative partner(s) can be helpful when finding and correcting errors.	**Marking the text** Provide students with program code that contains syntax and logic errors, such as missing statements or statements that are out of order. Have students use a highlighter to identify any syntax errors and then use arrows to rearrange the order of statements. Any missing statements should be written into the code.

© 2023 College Board

Practice 5: *Computing Innovations*

Skills	Tasks/Questions	Instructional Notes	Sample Activities and Strategies
5.A: Explain how computing systems work.	Explain how computing devices are networked and connected to the Internet to share data. **Key questions for students:** 1. What is the difference between the Internet and the World Wide Web? 2. How does the system handle a packet not arriving at its destination? 3. What would happen to the data if a path was no longer available, maybe because a line was severed?	Using the Internet is not the same as understanding how it works. Students need to be given opportunities to investigate multiple ways computers share data over networks.	**Kinesthetic learning** Simulate how messages are sent via the Internet by passing out index cards or other objects with messages on them. To simulate packets, divide the message up and write pieces of it on several separate index cards and number them. Pass the partial messages to students one at a time to ultimately get the complete message to an assigned destination (a preselected student in the room who will display the messages in the order they are received). Students can only hold one message in each hand and can only pass their message to a student with a free hand. Once all the messages have reached their destination, that student will reassemble the message and report it to the rest of the class. Be sure to debrief with students, pointing out that the packets could've taken different routes to get to the destination or arrived out of order.

continued on next page

© 2023 College Board

Practice 5: *Computing Innovations (cont'd)*

Skills	Tasks/Questions	Instructional Notes	Sample Activities and Strategies
5.B: Explain how knowledge can be generated from data.	Given categories of data and/or metadata, explain what information can be extracted through manipulation of those data. **Key questions for students:** 1. Would a different representation of the data help when analyzing them? 2. Based on the data and metadata available, what questions might you be able to answer? Is there additional information you might need?	If the data set provided is relatively small—for example, five items—students often create solutions that are equally limited and will only work with this specific set of data. Even if the data set is small, challenge students to create more abstract solutions that can be adapted to a larger data set.	**Activating prior knowledge** Before providing instruction on how information can be extracted from data, have students write down what they already know about computing devices with sensors (e.g., cellphones, voice assistants, fitness devices) and how these devices collect data. Have students consider how these data could be combined to create assumptions about the user.
5.C: Describe the impact of a computing innovation.	Given a passage about a computing innovation, describe potential beneficial and harmful effects of that computing innovation on society, the economy, or our culture. **Key questions for students:** 1. What are some ways a computing innovation has been used that weren't intended by the innovator? 2. What are some beneficial effects this computing innovation might have on you? What are some harmful effects? 3. How might a computing innovation contribute to the digital divide?	Before being able to describe the impact of a computing innovation, students need an understanding of what such an impact is and how it can be determined. This requires multiple opportunities for guided practice with feedback throughout the year.	**Discussion group** Have students research causes of the digital divide. During the class discussion, have students propose solutions to combat the digital divide. For each proposed solution, students should discuss what digital divide cause might be addressed, what barriers/obstacles the solution might pose, and any potential positive and negative impacts of the solution.

continued on next page

© 2023 College Board

Practice 5: *Computing Innovations (cont'd)*

Skills	Tasks/Questions	Instructional Notes	Sample Activities and Strategies
5.D: Describe the impact of gathering data.	Given a scenario related to data or a passage about a computing innovation, describe the potential risks to data privacy, storage, and security, as well as potential challenges that arise when processing data. When provided with larger data sets that may have poor formatting and missing data, identify the potential challenges and what might need to be done before using the data set. **Key questions for students:** 1. How can you work to combat bias in your data? 2. What are some ways in which, beyond their intended purpose, some computing innovations might be using user data? 3. What might need to be done to the data prior to being able to use them to answer a question?	When asked about data concerns, students often first think of hacking; however, there are other ways that data privacy can be compromised. Additional examples will help broaden students' thinking about what else might be a data concern for them or others.	**Marking the text** Provide students with the opportunity to mimic the process of cleaning data with a subset of raw data. Ask them to underline all text that has the same meaning but is represented differently (e.g., "street," "st," or "Street" would be equivalent). Next, have them circle areas of incomplete or invalid data.
5.E: Evaluate the use of computing based on legal and ethical factors.	Given scenarios related to computer usage, explain: ▪ Any legal and ethical concerns ▪ Any bias that might exist due to the way the computing innovation was developed or the way in which it is being used ▪ How computing resources can be misused, and unauthorized access obtained **Key questions for students:** 1. What can you do to protect your computing resources? 2. What are some ways that people might try to access your computing resources without your permission?	The way in which students use computers today can have future impacts on them or others. By using real-world examples, students can learn the consequences of failing to protect themselves and their data. With so much information readily available and seemingly free to use, legal ramifications and ethical concerns that may be associated with the use of information are not always intuitive to students.	**Look for a pattern** Challenge students to create a flawed data set of pictures that supports an incorrect claim. For example, a data set containing only pictures of red birds could lead one to believe that all birds are red. Have students share their data sets with the class and see if they can guess the flawed pattern. Afterwards, discuss how bias in data can result in biased computing innovations. Have them discuss how bias can show up in other data sources, such as text.

© 2023 College Board

Practice 6: *Responsible Computing*

Skills	Tasks/Questions	Instructional Notes	Sample Activities and Strategies
6.A: Collaborate in the development of solutions.	Move beyond group work, where work is simply divided among group members, to a collaborative process that involves respectful and mutual sharing of ideas, compromise, and an emphasis on conflict resolution. **Key questions for students:** 1. Are you collaborating with people who have different perspectives than you do? Is there an important perspective missing that you will need to research? 2. During your planning, how are you ensuring that all voices and opinions are being heard? 3. How will you make a decision when you have differing opinions?	Assigning group work is different from building the skills necessary for effective collaboration. Collaboration skills, such as consensus-building and conflict resolution, need to be explicitly taught and practiced, with feedback, to be effective. While collaboration on ideation and program design and development during the Create performance task is not required, it is encouraged.	**Pair programming** Introduce students to the concept of a text-based "choose-your-own-adventure" game where the user's choices determine the output and action in the program. Then, have students work in pairs, taking turns writing possible paths or outcomes for the character.
6.B: Use safe and secure methods when using computing devices.	Take steps to keep personal identifiable information from being shared while using computing devices. Be prudent about what is shared online. **Key questions for students:** 1. What steps are you taking to keep your identity safe when online? 2. How are you preventing access by others to your personal devices? 3. How do you decide what you will share publicly online? What are some of the consequences of sharing information about yourself online?	By having ongoing discussions regarding what data are being shared online about a student, with or without permission, students can make informed decisions about what to share publicly about themselves. Information that is put online is incredibly difficult, if not impossible, to remove.	**Discussion group** At the start of class, work with students to develop group norms around how to use the computing resources available in your school. Some norms might be using strong passwords, or establishing how data are shared among students during times of collaboration. As the year progresses, expand this list of norms to include topics outside the classroom as well, such as asking someone's permission before posting their picture on social media.

continued on next page

© 2023 College Board

Practice 6: *Responsible Computing (cont'd)*

Skills	Tasks/Questions	Instructional Notes	Sample Activities and Strategies
6.C: Acknowledge the intellectual property of others.	Provide citation for any media or program code that is being used in the development of a program that has come from someone other than the student or their collaborative partner(s). **Key questions for students:** 1. When are you required to credit another author, and what is the best way to do so? 2. What is the best way to give credit for open-source program code?	Some programming environments do not support the addition of comments into the program code. In these cases, comments can be added to a supporting document to acknowledge what work is not original. When completing the Create performance task, students will need to add substantial revisions and additional functionality when starting with preexisting program code. Simply changing an image in a game or the names of variables is insufficient.	**Sharing and responding** Have students anonymously share whether they think it is acceptable to listen to music or watch movies that have been downloaded illegally. Have a class discussion in which students respond to what was shared and discuss the real-life impacts and consequences of such actions. Have students brainstorm ways to protect the intellectual property of others.

© 2023 College Board

Using Strategies for Collaboration

Collaboration allows computer scientists to improve their products. Throughout the school year, students should be encouraged to collaborate with many different students, especially those who have different perspectives than their own. Collaborating with people who have a different perspective helps to reveal blind spots they might have due to their own backgrounds and experiences. Collaboration takes place in a variety of ways:

- Brainstorming ideas and solutions from different perspectives in a team environment
- Working together to design subtasks of a larger project, developing these subtasks, and then integrating them in the completion of the project
- Providing feedback on program design and development to improve the overall quality
- Providing technical support when problems arise that an individual is struggling to solve on their own

The cooperative learning strategies that are outlined in the instructional strategies table earlier can be used to help foster collaborative relationships in the classroom. Having an established protocol for students to have equal participation and share their ideas, such as the **think-pair-share** or **pair programming strategies**, builds students' confidence and can create a successful collaborative learning community within a classroom.

One way to assess how well students collaborate is by having them write reflections in a **journal** or participate in discussions about how collaborating with a partner helped accomplish their task or improve the quality of their program. Students should be asked to reflect not only on what went well in the collaboration process but also where they could improve it. Some partnerships require more intervention from the teacher than others. When a partnership is not working well, mediation and reflection can be helpful in getting it back on track. The reflection should focus on a student's own actions and how he or she works with others to solve problems. Teachers should guide students to share their reflections and consider different ways their work can be improved as they continue to collaborate.

© 2023 College Board

Differentiating Computer Science Instruction

With the availability of so many online resources about programming and other computer science concepts, some students may be self-taught in some of the introductory concepts for this course. While it is not required, students may have taken other introductory computer science courses prior to taking AP Computer Science Principles, depending on your school district. This varying level of computer science background and ability can make teaching the course challenging and the need for differentiated instruction greater. To engage all students in your classroom, consider trying the following:

- Provide leveled assignments in which students have a base project that demonstrates proficiency but can earn more points by completing additional challenges that move them to a more advanced demonstration of their understanding.
- Open-ended projects allow students to challenge themselves at their individual ability levels and interests.
- Pair programming is a way for students to support each other when developing a program.
- Give students extended projects or reading assignments that can be completed over the course of a week. These projects and assignments can be homework for students who need the class time to complete longer projects that might require assistance from the teachers, while students who are fast finishers will still have something engaging and class-related to work on. One example might be assignments from online practice sites, such as **CodingBat** and **Khan Academy**. A minimum set of problems can be assigned with incentives for students who are eager to learn more.
- Some students may need specific accommodations or additional support when taking AP Computer Science Principles. College Board fully expects that students with disabilities will receive all of the aids and services called for in their school plans. However, additional assistance or devices may be useful to fully access this course. For example, students who are physically disabled, blind, or visually impaired might need alternate formats for activities. They may also require special equipment, such as screen readers. Block-based programming languages can be more difficult for these students, so the use of a specially designed text-based programming environment, such as **Quorum**, might be more suitable. For more information on how to make your computer science classroom more accessible to students with disabilities, we recommend reviewing **Access Computing** resources.

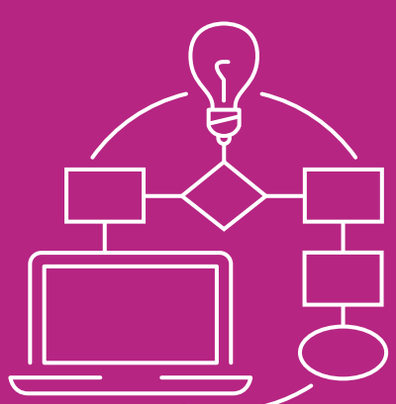
The goal is to ensure that all students are engaged in learning computer science. Students with some experience or who pick up on material quickly will need incremental challenges, while students with less experience may need more time and support to accomplish tasks.

© 2023 College Board

THIS PAGE IS INTENTIONALLY LEFT BLANK.

AP COMPUTER SCIENCE PRINCIPLES

Curriculum Alignment



Curriculum Alignment

Each AP endorsed provider offers a complete curriculum and syllabus for teachers to adopt. Teachers can also choose to create their own curriculum and syllabus based on the AP Computer Science Principles course framework. Because AP Computer Science Principles is presented as big ideas instead of units, teachers who choose to create their own curriculum and syllabus will need to develop their own unit plans to scaffold and integrate the big ideas and skills in this course.

To aid in the sequencing of your course and assigning the Topic Questions from **AP Classroom**, we have provided a blank Unit at a Glance page. This page can be replicated as you plan your units and align the topics you intend to cover in that unit. It is important to note that the big ideas in this course are themes that should be spiraled throughout the course, rather than taught in isolation as units. When creating a unit plan, it is common for teachers to pair content from multiple big ideas and skills.

While we have segmented each big idea into topics, there may be times when you introduce part of a topic in one unit and revisit it one or more times in future units. In these cases, there may be some Topic Questions that students can answer during the initial introduction of the topic, while other questions would be more appropriate to use later in the school year. You can preview questions in AP Classroom before assigning them to students. When pacing out the number of class periods to spend on each topic, be sure to include time for students to complete the corresponding Topic Questions.

If you choose an AP endorsed provider, check to see if they have developed their own Unit at a Glance chart or something similar that can be printed and added to this section. This document should align the provider curriculum to the AP Computer Science Principles Course and Exam Description *and indicate where it may be appropriate to assign formative Topic Questions in* ***AP Classroom***.

© 2023 College Board

Unit at a Glance

Unit _____ : _____

This page provides a place to plan instruction and unit pacing for this course. When planning, be sure to:

- Select topics from multiple big ideas to create units.
- Select activities and lessons that best facilitate students' learning and practice of the required content and skills.
- Schedule time for students to complete the Topic Questions from **AP Classroom** at home or in class.
- Meet the curricular requirement by including at least three investigations into computing innovations and covering the beneficial and harmful effects, how the innovation uses data, and any privacy, security, or storage concerns.
- Allow **9 hours in class** for students to complete the Create performance task.

Topic	Learning Objectives and Skills	Topic Questions	Class Periods	Instructional Planning/Resources

© 2023 College Board

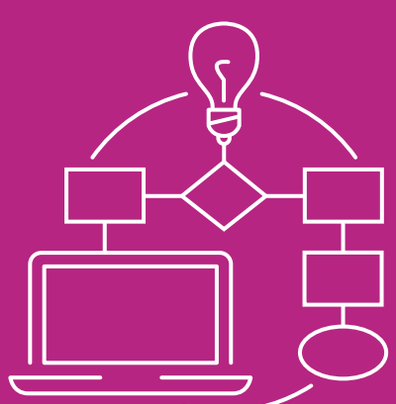
Topic	Learning Objectives and Skills	Topic Questions	Class Periods	Instructional Planning/Resources

© 2023 College Board

THIS PAGE IS INTENTIONALLY LEFT BLANK.

AP COMPUTER SCIENCE PRINCIPLES

Exam Information



Exam Overview

The AP Computer Science Principles Exam assesses student understanding of the computational thinking practices and learning objectives outlined in the course framework. The exam consists of the Create performance task and an end-of-course AP Exam. The Create performance task requires at least 9 hours of dedicated class time for students to complete. The end-of-course exam is 3 hours long and includes 70 multiple-choice questions and 2 written response questions related to their Create performance task. The two questions include four distinct prompts: Written Response 1, Written Response 2(a), Written Response 2(b), and Written Response 2(c). Students will be given their student-authored Personalized Project Reference to use when answering these written response prompts.

The multiple-choice section will include three different types of questions (in the following order on the exam): single-select questions, single-select questions with a reading passage about a computing innovation, and multi-select questions. As part of the exam, students will be given the Exam Reference Sheet (see Appendix), which contains both block-based and text-based programming constructs and establishes a common way to communicate programming concepts for the purpose of the exam. The details of the exam, including weighting and timing, can be found below:

© 2023 College Board

Section	Question Type/Component	Number of Questions	Exam Weighting	Timing
I	Multiple-choice questions	70	70%	120 minutes End-of-course AP Exam
	Single-select	57		
	Single-select with reading passage about a computing innovation	5		
	Multi-select	8		
II	Create Performance Task	*See Below*	30%	*See Below*
	Program code, video, and Personalized Project Reference			At least 9 hours in class
	Written response questions related to the Create performance task	2		60 minutes End-of-course AP Exam

The AP Exam assesses each of the five big ideas of the course with the following weighting on the multiple-choice section:

Big Ideas	Exam Weighting
Big Idea 1: Creative Development	**10–13%**
Big Idea 2: Data	**17–22%**
Big Idea 3: Algorithms and Programming	**30–35%**
Big Idea 4: Computer Systems and Networks	**11–15%**
Big Idea 5: Impact of Computing	**21–26%**

Questions in Big Ideas 1, 2, and 3 can be represented as algorithms (with no program code) or as program code using the Exam Reference Sheet (see Appendix). The program code questions will contain some graphical representations, some of which use robots in a grid.

© 2023 College Board

How Student Learning Is Assessed on the AP Exam

The AP Computer Science Principles computational thinking practices are assessed on the AP Exam as detailed below.

Section I: Multiple Choice

The AP Computer Science Principles Exam multiple-choice section has 70 total questions, including 65 individual questions and one set of five questions that uses a reading passage about a computing innovation as a stimulus.

All computational thinking practices except Computational Thinking Practice 6 are assessed in the multiple-choice section, with the following exam weighting:

Computational Thinking Practice	Exam Weighting
Practice 1: Computational Solution Design	**18–25%**
Practice 2: Algorithms and Program Development	**20–28%**
Practice 3: Abstraction in Program Development	**7–12%**
Practice 4: Code Analysis	**12–19%**
Practice 5: Computing Innovations	**28–33%**

SINGLE-SELECT QUESTIONS WITH READING PASSAGE

The set of five questions associated with a reading passage about a computing innovation assess Computational Thinking Practices 3 and 5. The following is an example reading passage.

Reading Passage

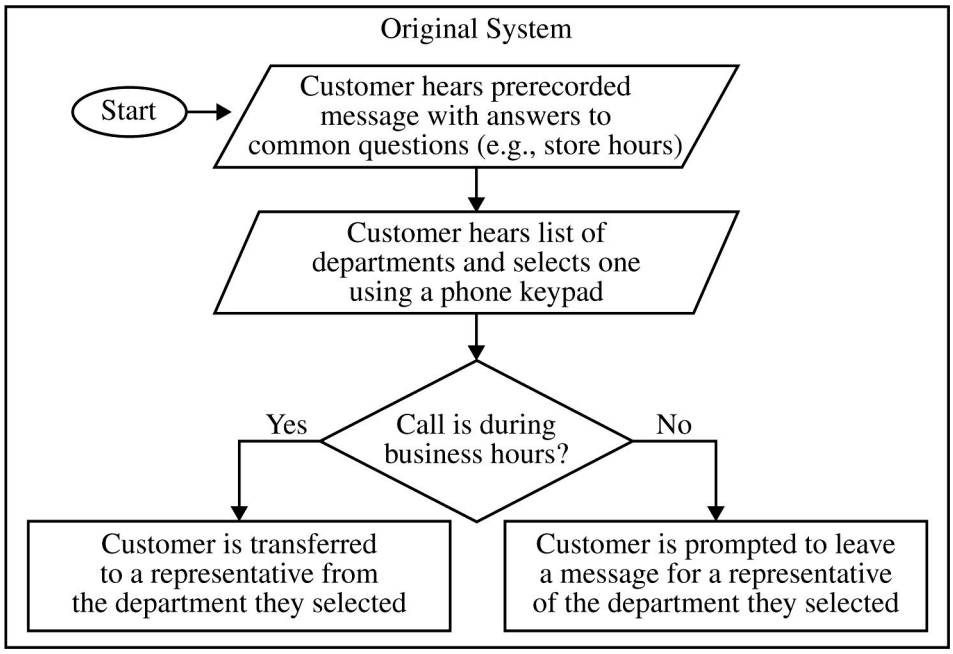
A chain of retail stores uses software to manage telephone calls from customers. The system was recently upgraded. Customers interacted with the original system using their phone keypad. Customers interact with the upgraded system using their voice.

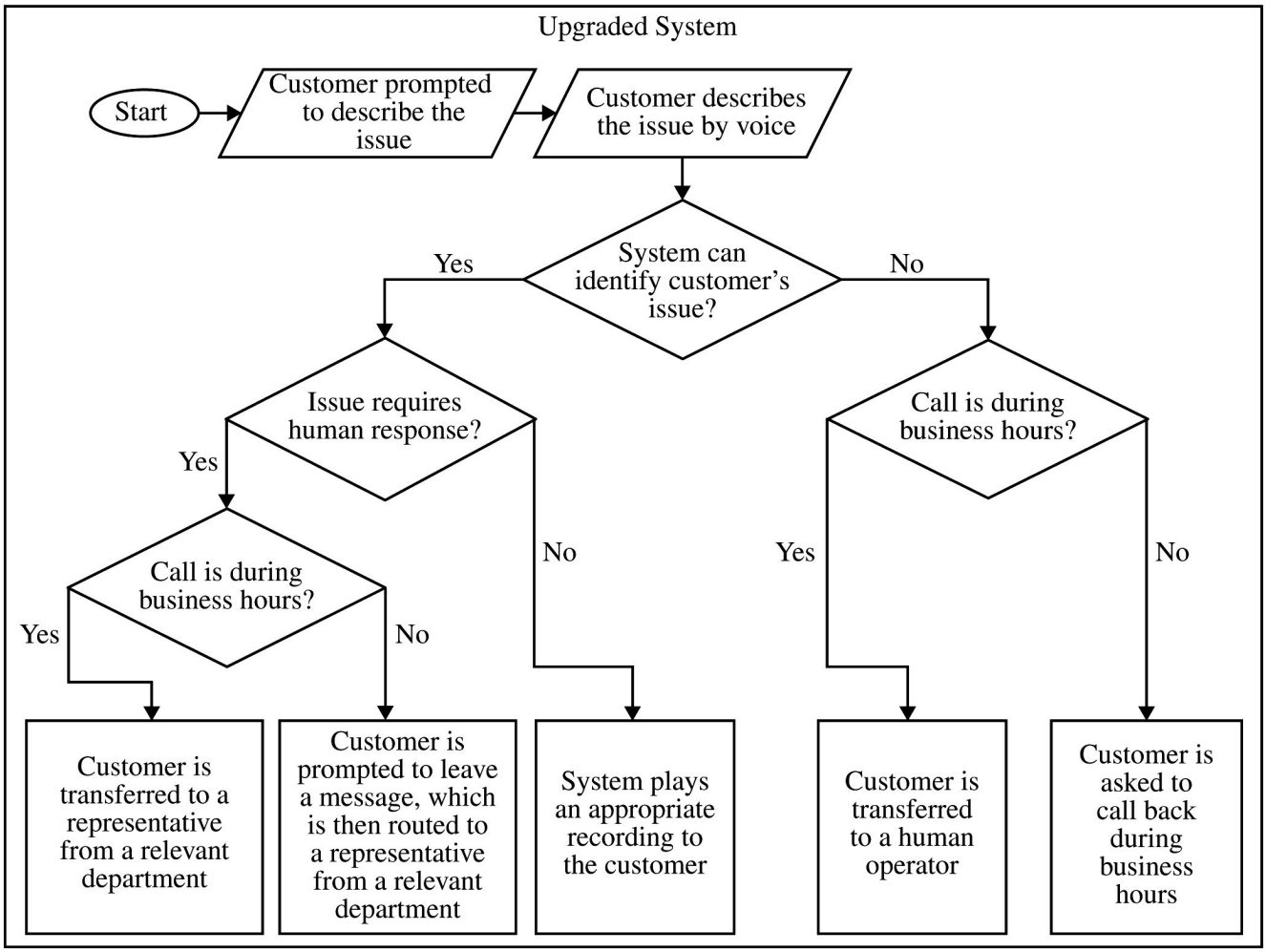
The upgraded system (but not the original system) stores all information from the calling session in a database for future reference. This includes the customer's telephone number and any information provided by the customer (name, address, order number, credit card number, etc.).

The original system and the upgraded system are described in the following flowcharts. Each flowchart uses the following blocks.

Block	Explanation
Oval	The start of the algorithm
Parallelogram	An input or output step
Diamond	A conditional or decision step, where execution proceeds to the side labeled "Yes" if the answer to the question is yes and to the side labeled "No" if the answer to the question is no
Rectangle	The result of the algorithm

© 2023 College Board





© 2023 College Board

All reading passage questions will be worded in a similar way. The following are examples of types of questions associated with reading passage questions; note that only five of these examples would be included with any one reading passage. The text in italics will vary by question, while the remainder of the prompt will be consistently used in all reading passage questions. This *Call Center* reading passage and a subset of its associated set of questions can be found in the Sample Exam Questions section starting on page 172. A full set of five questions for the *Call Center* reading passage can be found in the question bank in AP Classroom.

Practice 3: Abstraction in Program Development

- Which of the following input data [must be obtained/is needed] *by the upgraded system that was NOT needed by the original system*?
- Which of the following data is not [obtained/provided] *directly from the user* but is necessary for *the upgraded system to operate as described*?
- Which of the following data is necessary for the *Call Center* to process in order to enable *it to provide an answer to the caller?*
- Which of the following is [LEAST/MOST] likely to be included in *the directory*?

Practice 5: Computing Innovations

- Which of the following is considered a potential effect of *the application* rather than a [function/purpose] *of the application*?
- Which of the following is [LEAST/MOST] likely to be a [benefit/harm] of *storing the information from each calling session in a database*?
- Of the following potential benefits, which is [LEAST/MOST] likely to be provided by *the upgraded system?*
- Which of the following may be an unintended effect of the use of *Call Center*?
- Which of the following is the [LEAST/MOST] [likely/plausible] data [privacy/security/storage] concern *of the upgraded system*?
- Which of the following *groups* is [LEAST/MOST] likely to receive *targeted advertisements*?
- Which of the following statements is [LEAST/MOST] likely to be true about the tradeoffs of *the Call Center recording the caller's phone number?*

Section II: Create Performance Task

The second section of the AP Computer Science Principles Exam consists of a through-course performance task where students will develop a computer program of their choice and a written response section based on the performance task that is completed during the end-of-course AP exam. Students will be provided 9 hours of in-class time to complete their program, video, and Personalized Project Reference.

On the end-of-course AP exam, students will respond to four distinct prompts, Written Response 1, Written Response 2(a), Written Response 2(b), and Written Response 2(c), related to the code contained in the student-authored Personalize Project Reference. Students will have access to this reference while responding to these prompts. The four categories listed below align to the four prompts.

© 2023 College Board

Prompt Category	Related Learning Objectives
Written Response 1: Program Design, Function, and Purpose	Students should be prepared to respond to prompts about their program that assess any of the following learning objectives: ▪ CRD-2.A: Describe the purpose of a computing innovation. ▪ CRD-2.B: Explain how a program or code segment functions. ▪ CRD-2.C: Identify input(s) to a program. ▪ CRD-2.D: Identify output(s) produced by a program. ▪ CRD-2.E: Develop a program using a development process. ▪ CRD-2.F: Design a program and its user interface. ▪ CRD-2.G: Describe the purpose of a code segment or program by writing documentation.
Written Response 2(a): Algorithm Development	Students should be prepared to respond to prompts about their program that assess any of the following learning objectives: ▪ CRD-2.B: Explain how a program or code segment functions. ▪ AAP-2.E.b: Evaluate expressions that use relational operators. ▪ AAP-2.F.b: Evaluate expressions that use logic operators. ▪ AAP-2.H.b: Determine the result of conditional statements. ▪ AAP-2.J: Express an algorithm that uses iteration without using a programming language. ▪ AAP-2.K.b: Determine the result or side effect of iteration statements. ▪ AAP-2.L: Compare multiple algorithms to determine if they yield the same side effect or result. ▪ AAP-2.M.a: Create algorithms. ▪ AAP-2.M.b: Combine and modify existing algorithms.
Written Response 2(b): Errors and Testing	Students should be prepared to respond to prompts about their program that assess any of the following learning objectives: ▪ CRD-2.I.a: Identify the error. ▪ CRD-2.I.b: Correct the error. ▪ CRD-2.J: Identify inputs and corresponding expected outputs or behaviors that can be used to check the correctness of an algorithm or program.
Written Response 2(c): Data and Procedural Abstraction	Students should be prepared to respond to prompts about their program that assess any of the following learning objectives: ▪ AAP-1.D.a: Develop data abstraction using lists to store multiple elements. ▪ AAP-1.D.b: Explain how the use of data abstraction manages complexity in program code. ▪ AAP-2.O.a: Write iteration statements to traverse a list. ▪ AAP-2.O.b: Determine the result of an algorithm that includes list traversals. ▪ AAP-3.B: Explain how the use of procedural abstraction manages complexity in a program.

© 2023 College Board

The second section of the AP Computer Science Principles Exam assesses Computational Thinking Practices 1, 2, 3, and 4 across six rubric rows.

- **Course Project: Video** assesses students' ability to implement and apply an algorithm. (Skill 2.B)
- **Course Project: Program Requirements** assesses students' ability to implement and apply an algorithm. (Skill 2.B)
- **Written Response 1: Program Design, Function, and Purpose** assesses students' ability to investigate the situation, context, or task. (Skill 1.A)
- **Written Response 2(a): Algorithm Development** assesses students' ability to explain how a code segment of program functions. (Skill 4.A)
- **Written Response 2(b): Errors and Testing** assesses students' ability to identify and correct errors in algorithms and programs, including error discovery through testing. (Skill 4.C)
- **Written Response 2(c): Data and Procedural Abstraction** assesses students' ability to explain how abstraction manages complexity. (Skill 3.C)

© 2023 College Board

Performance Task Verbs

The following task verbs are commonly used in the performance task and written responses:

Capture: Select a portion of program code that addresses the prompt(s).

Demonstrate: Provide sufficient evidence for an answer or point being made.

Describe: Provide the relevant features or characteristics of what the program code represents or is being used to accomplish.

Design: Develop a plan for how to accomplish the program specification or requirements.

Explain: Provide information about how or why a relationship, situation, or outcome occurs, listing detailed steps of the algorithm or using evidence and/or reasoning.

Identify: Provide a name for the specific topic, without elaboration or explanation.

Implement/Write: Recognize and use proper syntax to execute the program design.

© 2023 College Board

Preparing for the Create Performance Task

Academic Integrity and Plagiarism Policy

This policy addresses plagiarism and academic integrity in completing the Create Performance Task.

Plagiarism

The use of program code, media (e.g., video, images, sound), data, information, or evidence created by someone else or with generative AI tools in the creation of a program and/or a program code segment(s), without appropriate acknowledgment (i.e., through citation, through attribution, and/or by reference), is considered **plagiarism**. A student who commits plagiarism will receive a score of 0 on the Create performance task, including their responses to the written response prompts on the end-of-course AP Exam.

To the best of their ability, teachers will ensure that students understand how to ethically incorporate ideas that are not their own and provide credit to the original creator or source, as well as the consequences of plagiarism.

Acceptable Generative AI Use

Students are permitted to utilize generative AI tools as supplementary resources for understanding coding principles, assisting in code development, and debugging. This responsible use aligns with current guidelines for peer collaboration on developing code.

Students should be aware that generative AI tools can produce incomplete code, code that creates or introduces biases, code with errors, inefficiencies in how the code executes, or code complexities that make it difficult to understand and therefore explain the code. It is the student's responsibility to review and understand any code co-written with AI tools, ensuring its functionality. Additionally, students must be prepared to explain their code in detail, as required on the end-of-course AP Exam.

Preparing for Final Submission

Students are not permitted to collaborate on the video or creation of the Personalized Project Reference.

The Personalized Project Reference cannot include course content or comments within the code or on any other part of the reference. Including course content or comments in the Personalized Project Reference will result in students receiving a score of 0 on the Create performance task, including their responses to the written response prompts on the end-of-course AP Exam.

Attestations

During the final submission process in the AP Digital Portfolio, students will be asked to attest that they have followed the Performance Task guidelines and have not plagiarized their submission. Each of the three components of the Create performance

© 2023 College Board

task must be submitted as final to be sent for scoring. Additionally, if students do not submit their Personalized Project Reference by the deadline, they will not have this resource available on exam day to complete their written response section.

Preparing Students for the Task

*Prior to students beginning a performance task, teachers **should** do the following with students:*

- Provide instruction, practice, and feedback related to content and skills that will help students succeed on the performance task. This can include, but needs not be limited to, the iterative development process, strategies for collaboration, the development of both data and procedural abstractions, describing an algorithm's purpose and explaining how it functions, and identifying test data that demonstrates the different outcomes of an algorithm.
- Brainstorm problems that programming can address or brainstorm special interests that students want to incorporate when developing a program.
- Assist students in defining their focus and choice of topics without making selections for them (e.g., by asking questions).
- Review the performance task directions and provide multiple opportunities to practice and discuss all or part of the performance task. Formative performance task prompts can be assigned in **AP Classroom**.
- Explain the role the teacher can and cannot play in providing students with assistance during the actual performance task; teachers should encourage students to take advantage of the opportunity to get assistance and feedback from them during practice.
- Inform students that they cannot submit any work from AP Central samples, curriculum provider samples, or practice performance tasks for their final submission.
- Provide guidance to students about how they can use media or data sources in their program code and how to avoid plagiarism when doing so. Any media or data sources that have not been created by the student must be acknowledged, and credit must be given to the author.
- Provide guidance to students about how they can integrate existing program code into their own program and extend it in some new way by adding new functionality. Any program code that has not been written by the student—including starter code provided by you as their teacher, code found in APIs or open-source code, and code generated using AI tools—should be acknowledged, and credit should be given to the author. Some examples of ways to provide attribution for program code that was not authored by the student are as follows:
  - If the program code has been made available for use by their teacher, students should add a comment that states: This code was provided as starter code by my teacher.
  - If the program code has been made available through an API or open-source code, students should add a comment that states: This code was made freely available by [source of code].
  - If the program code has been co-created with the assistance of a generative AI tool, students should add a comment that states: This code was generated using [Generative AI Tool Name].
- Instruct students how to add comments to their program code to acknowledge and credit authors of program code, media, or data sources.
  - If the programming environment allows students to include comments, this is the preferred way to acknowledge and give credit to another author.

© 2023 College Board

- If the programming environment does not allow students to include comments, students can add comments in a document editor when they capture their program code for submission to acknowledge and give credit to another author.

- Provide explicit instructions about the AP Digital Portfolio file submission requirements and process to ensure that students' work is sent for scoring. NOTE: All three components must be submitted as final by the deadline posted on AP Central. Only files that are submitted as final will be sent for scoring. Additionally, if students do not submit their Personalized Project Reference by the deadline, they will not have this resource available on exam day to complete their written response section.
- Encourage students to keep a programming journal of the design choices made during the development of the program code or code segment and the effect of these decisions on the program's function.

Administering the Performance Task

*Once students have started their official administration of the performance task, teachers **must**:*

- Provide a minimum of 9 hours of class time to complete this task.
- Inform students of the final submission date and time, which can be found on the **AP Computer Science Principles Exam page**. Remind students that all three components must be submitted as final. Only files that are submitted as final will be sent for scoring. Additionally, if students do not submit their Personalized Project Reference by the deadline, they will not have this resource available on exam day to complete their written response section.
- Carefully plan a calendar that provides time for all performance task components to be completed and uploaded in advance of the deadline.
- Ensure that students are aware of the performance task directions (found in the Student Handouts section).
- Allow students to collaborate only during the ideation and development, including debugging and error testing, of the program code, if they choose to do so. **NOTE: students are not allowed to collaborate on their video or their Personalized Project Reference**.
- Allow students' interest to drive their choice of projects and programming languages. Encourage students to use a programming language they know well. The administration of the Create performance task is not the time to learn a new programming language.
- Assist in resolving technical problems that impede work, such as a failing workstation or difficulty with access to networks, or to help with saving files.
- Wait until after all three components of the students' performance tasks have been completed and submitted as final to the AP Digital Portfolio before providing feedback on those components and any practice written responses (if they are being considered as part of the class grade).
- Advise students that they may not revise their work once they have completed and submitted it as final to the AP Digital Portfolio. This includes their Personalized Project Reference. It is advisable that students practice at least one sample set of written response prompts to check for themselves that the code included in the Personalized Project Reference will be sufficient to respond to these prompts. Teachers are not allowed to provide any feedback on the written response prompts or any components of the Create performance task until after all three components are submitted as final to the AP Digital Portfolio.
- Advise students that the screen captures of their program code provided in the Personalized Project Reference cannot include any course content beyond the program code. All comments that were included in the program code during the

© 2023 College Board

development of their program should be removed prior to taking screen captures. Including course content or comments in the screen captures will result in the written response portion of the student's exam being scored a 0.

- Instruct students how to capture their program code to submit for the performance task.
  - With text-based program code, students can use the print command to save their program code as a PDF file, or they can copy and paste their code to a text document and then convert it into a PDF file.
  - With block-based program code, students can create screen captures that include only their program code, paste these images into a document, and then convert that document to a PDF file. Screen captures should not be blurry, and text should be at least 10 pt font size.
  - For the Personalized Project Reference, screen captures should not be blurry, and text should be at least 10 pt font size. Any course content or comments that were included with the program code, should not be included in the screen captures included in the Personalized Project Reference. Including course content or comments in the Personalized Project Reference could result in students receiving a 0 for the Create performance task.
- Share the categories for each written response question on the end-of-course AP Exam. The written response questions will be related to the student's Create performance task.
- Review the final submitted files for each performance task component. These files should be returned to students if they are the incorrect file, are corrupt, not readable, or there is course content or comments in the Personalized Project Reference. Teachers **may not** return a file to a student due to the quality of the work submitted.

*Once students have started their official administration of the performance task, teachers **may not**:*

- Assign, provide, or distribute to students specific topics or a program to develop.
- Write, revise, amend, or correct student work, including debugging the program, writing or designing functionality in the program, testing the program, or making revisions to the program.
- Allow students to submit practice performance tasks for AP assessment scoring.
- Allow students to collaborate during the creation of their video or their Personalized Project Reference.
- Provide feedback on practice written response prompts before all three components of the Create performance task have been submitted as final in the AP Digital Portfolio.

*Once students have started their official administration of the performance task, teachers **may**:*

- Oversee the formation of groups.
- Clarify the requirements for the performance task when it is clear students do not understand the directions.
- Designate consecutive or nonconsecutive class hours to complete the performance task.
- Continue whole-class teaching of course content and skills during time not designated to complete the performance task.
- Resolve collaboration issues when one collaborative partner is clearly and directly impeding the completion of the performance task.
- Inform students that the scoring process that occurs in the AP Reading is different from the one that may be used in the classroom; the AP score that students receive may be different than their classroom grade.

© 2023 College Board

Sample Exam Questions

The sample exam questions that follow illustrate the relationship between the course framework and AP Computer Science Principles Exam and serve as examples of the types of questions that appear on the exam. After the sample questions, you will find a table that shows the skill and learning objective(s) to which each question relates. The table also provides the answers to the multiple-choice questions.

Section I: Multiple Choice

1. Which of the following best explains how data is typically assembled in packets for transmission over the Internet?

   (A) Each packet contains data to be transmitted, along with metadata containing information used for routing the data.

   (B) Each packet contains an encrypted version of the data to be transmitted, along with metadata containing the key needed to decrypt the data.

   (C) Each packet contains only the metadata used to establish a direct connection so that the data can be transmitted.

   (D) Each packet contains multiple data files bundled together, along with metadata describing how to categorize each data file.

2. Each student that enrolls at a school is assigned a unique ID number, which is stored as a binary number. The ID numbers increase sequentially by 1 with each newly enrolled student. If the ID number assigned to the last student who enrolled was the binary number 1001 0011, what binary number will be assigned to the next student who enrolls?

   (A) 1001 0100

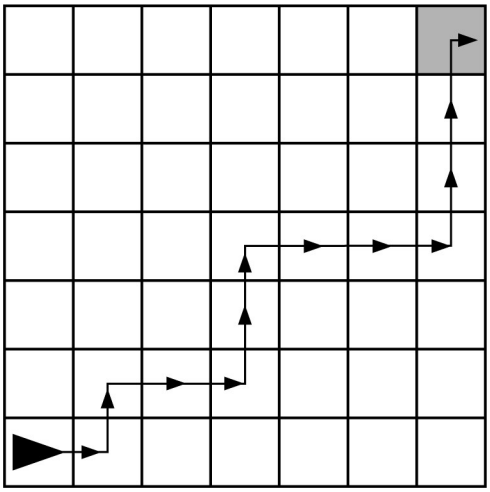
   (B) 1001 0111

   (C) 1101 0100

   (D) 1101 0111

© 2023 College Board

3. The following grid contains a robot represented as a triangle. The robot is initially facing right.



Which of the following code segments can be used to move the robot to the gray square along the path indicated by the arrows?

(A)

```
n ← 1
REPEAT 3 TIMES
    n ← n + 1
    REPEAT n TIMES
        MOVE_FORWARD
        ROTATE_LEFT
        MOVE_FORWARD
        ROTATE_RIGHT
```

(B)

```
n ← 1
REPEAT 3 TIMES
    REPEAT n TIMES
        MOVE_FORWARD
        ROTATE_LEFT
        MOVE_FORWARD
        ROTATE_RIGHT
    n ← n + 1
```

(C)

```
n ← 1
REPEAT 3 TIMES
    n ← n + 1
    REPEAT n TIMES
        MOVE_FORWARD
    ROTATE_LEFT
    REPEAT n TIMES
        MOVE_FORWARD
    ROTATE_RIGHT
```

(D)

```
n ← 1
REPEAT 3 TIMES
    REPEAT n TIMES
        MOVE_FORWARD
    ROTATE_LEFT
    REPEAT n TIMES
        MOVE_FORWARD
    ROTATE_RIGHT
    n ← n + 1
```


© 2023 College Board

4. A user purchased a new smart home device with embedded software and connected the device to a home network. The user then registered the device with the manufacturer, setting up an account using a personal e-mail and password. Which of the following explains how a phishing attack could occur against the user of the smart home device?

   (A) A vulnerability in the device's software is exploited to gain unauthorized access to other devices on the user's home network.

   (B) A vulnerability in the device's software is exploited to install software that reveals the user's password to an unauthorized individual.

   (C) The user is sent an e-mail appearing to be from the manufacturer, asking the user to confirm the account password by clicking on a link in the e-mail and entering the password on the resulting page.

   (D) The user's account is sent an overwhelming number of messages in an attempt to disrupt service on the user's home network.

5. Which of the following school policies is most likely to have a positive impact on the digital divide?

   (A) A school allows students to bring a graphing calculator from home to complete in-class mathematics assignments.

   (B) A school allows students to bring a tablet computer to class every day to participate in graded quizzes.

   (C) A school provides a laptop or tablet computer to all students enrolled at the school.

   (D) A school recommends that all students purchase a computer with as much processing speed as possible so that projects run faster.

6. In a certain country, a person must be at least 16 years old to drive a car and must be at least 18 years old to vote. The variable `age` represents the age of a person as an integer.

   Which of the following expressions evaluates to `true` if the person is old enough to drive but not old enough to vote, and evaluates to `false` otherwise?

   I. `(age ≥ 16) AND (age ≤ 18)`

   II. `(age ≥ 16) AND (NOT(age ≥ 18))`

   III. `(age < 18) AND (NOT(age < 16))`

   (A) II only

   (B) I and II only

   (C) I and III only

   (D) II and III only

© 2023 College Board

7. A Web site uses several strategies to prevent unauthorized individuals from accessing user accounts. Which of the following is NOT an example of multifactor authentication?

   (A) Each employee for a company is issued a USB device that contains a unique token code. To log into a company computer, an employee must insert the USB device into the computer and provide a correct password.

   (B) After logging into an account from a new device, a user must enter a code that is sent via e-mail to the e-mail address on file with the account.

   (C) In order to log into an account, a user must provide both a password and a fingerprint that is captured using the user's device.

   (D) When a user enters an incorrect password more than two times in a row, the user is locked out of the account for 24 hours.

8. A list of numbers has `n` elements, indexed from `1` to `n`. The following algorithm is intended to display the number of elements in the list that have a value greater than `100`. The algorithm uses the variables `count` and `position`. Steps 3 and 4 are missing.

   Step 1: Set `count` to `0` and `position` to `1`.
   Step 2: If the value of the element at index `position` is greater than `100`, increase the value of `count` by `1`.
   Step 3: (missing step)
   Step 4: (missing step)
   Step 5: Display the value of `count`.

   Which of the following could be used to replace steps 3 and 4 so that the algorithm works as intended?

   (A) Step 3: Increase the value of `position` by `1`.
   Step 4: Repeat steps 2 and 3 until the value of `count` is greater than `100`.

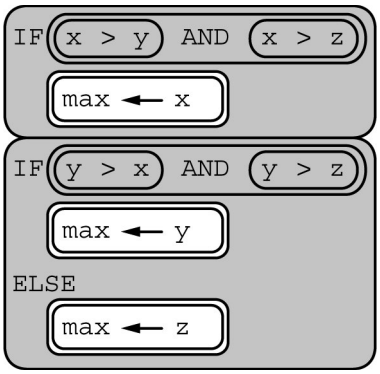
   (B) Step 3: Increase the value of `position` by `1`.
   Step 4: Repeat steps 2 and 3 until the value of `position` is greater than `n`.

   (C) Step 3: Repeat step 2 until the value of `count` is greater than `100`.
   Step 4: Increase the value of `position` by `1`.

   (D) Step 3: Repeat step 2 until the value of `position` is greater than `n`.
   Step 4: Increase the value of `count` by `1`.

© 2023 College Board

9. The following code segment is intended to set `max` equal to the maximum value among the integer variables `x`, `y`, and `z`. The code segment does not work as intended in all cases.



Which of the following initial values for `x`, `y`, and `z` can be used to show that the code segment does not work as intended?

(A) `x = 1, y = 2, z = 3`

(B) `x = 1, y = 3, z = 2`

(C) `x = 2, y = 3, z = 1`

(D) `x = 3, y = 2, z = 1`

10. A digital photo file contains data representing the level of red, green, and blue for each pixel in the photo. The file also contains metadata that describe the date and geographic location where the photo was taken. For which of the following goals would analyzing the metadata be more appropriate than analyzing the data?

(A) Determining the likelihood that the photo is a picture of the sky

(B) Determining the likelihood that the photo was taken at a particular public event

(C) Determining the number of people that appear in the photo

(D) Determining the usability of the photo for projection onto a particular color background

© 2023 College Board

11. The following procedure is intended to return the number of times the value `val` appears in the list `myList`. The procedure does not work as intended.

```
Line 1: PROCEDURE countNumOccurences(myList, val)
Line 2: {
Line 3:     FOR EACH item IN myList
Line 4:     {
Line 5:         count ← 0
Line 6:         IF(item = val)
Line 7:         {
Line 8:             count ← count + 1
Line 9:         }
Line 10:    }
Line 11:    RETURN(count)
Line 12:}
```

Which of the following changes can be made so that the procedure will work as intended?

(A) Changing line 6 to `IF(item = count)`

(B) Changing line 6 to `IF(myList[item] = val)`

(C) Moving the statement in line 5 so that it appears between lines 2 and 3

(D) Moving the statement in line 11 so that it appears between lines 9 and 10

12. A certain computer has two identical processors that are able to run in parallel. Each processor can run only one process at a time, and each process must be executed on a single processor. The following table indicates the amount of time it takes to execute each of three processes on a single processor. Assume that none of the processes are dependent on any of the other processes.

Process	Execution Time on Either Processor
X	60 seconds
Y	30 seconds
Z	50 seconds

Which of the following best approximates the minimum possible time to execute all three processes when the two processors are run in parallel?

(A) 60 seconds

(B) 70 seconds

(C) 80 seconds

(D) 90 seconds

© 2023 College Board

13. A sorted list of numbers contains 500 elements. Which of the following is closest to the maximum number of list elements that will be examined when performing a binary search for a value in the list?

(A) 10

(B) 50

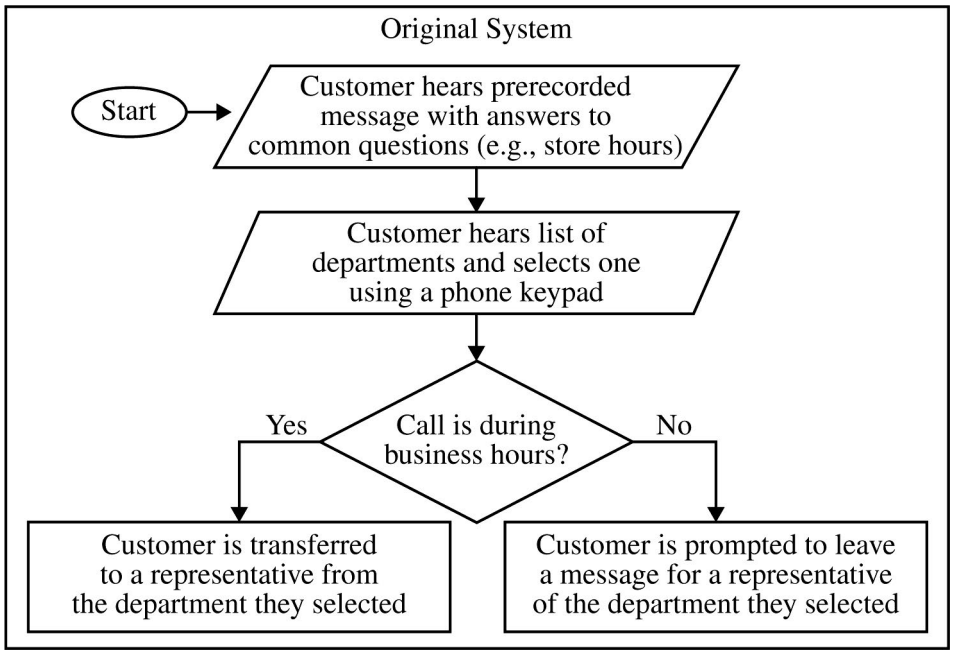
(C) 250

(D) 500

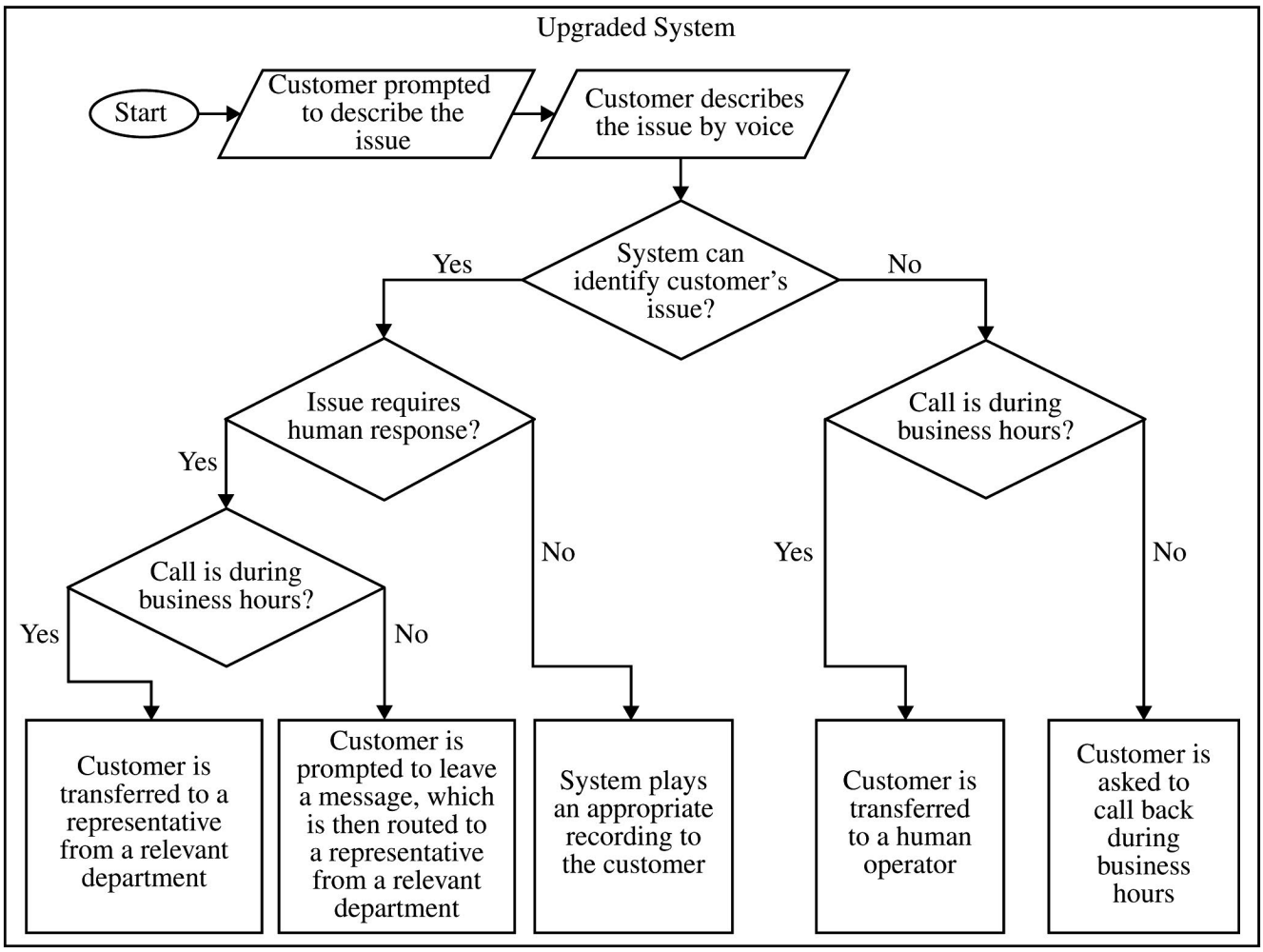
Questions 14–16 refer to the information below. Note: This is a representative sample of questions related to the reading passage. There will be five single-select multiple-choice questions related to a reading passage on the AP Exam.

- A chain of retail stores uses software to manage telephone calls from customers. The system was recently upgraded. Customers interacted with the original system using their phone keypad. Customers interact with the upgraded system using their voice.
- The upgraded system (but not the original system) stores all information from the calling session in a database for future reference. This includes the customer's telephone number and any information provided by the customer (name, address, order number, credit card number, etc.).
- The original system and the upgraded system are described in the following flowcharts. Each flowchart uses the following blocks.

Block	Explanation
Oval	The start of the algorithm
Parallelogram	An input or output step
Diamond	A conditional or decision step, where execution proceeds to the side labeled "Yes" if the answer to the question is yes and to the side labeled "No" if the answer to the question is no
Rectangle	The result of the algorithm



© 2023 College Board



14. The upgraded system uses a directory containing additional information not supplied by the customer. The directory is used to help direct calls effectively. Which of the following is LEAST likely to be included in the directory?

(A) A list of common issues and whether each issue requires a human representative

(B) A list of common keywords or phrases and a corresponding issue for each keyword or phrase

(C) A list of computers the company owns and the computers' corresponding IP addresses

(D) A list of human representatives and the corresponding department for each representative

© 2023 College Board

15. Of the following potential benefits, which is LEAST likely to be provided by the upgraded system?

    (A) Human representatives will not be needed to respond to some inquiries.

    (B) The company will be able to provide a human representative for any incoming call.

    (C) Customers are likely to spend less time listening to information not relevant to their issue.

    (D) Customers will be unable to mistakenly select the incorrect department for their particular issue.

16. Which of the following is the most likely data privacy concern of the upgraded system?

    (A) Customers' personal information could be compromised if an unauthorized individual gains access to the call session database.

    (B) Storing information in the call session database makes it easy for individuals to trick the system using malicious links.

    (C) The system design increases the chance that customers will unknowingly install malware on their devices that will share their data with unauthorized individuals.

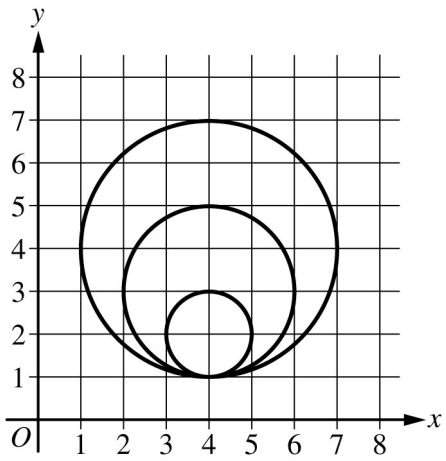
    (D) The system design makes it easy for unauthorized individuals to acquire customers' private encryption keys.

© 2023 College Board

17. Consider the following procedure.

Procedure Call	Explanation
`drawCircle(xPos, yPos, rad)`	Draws a circle on a coordinate grid with center `(xPos, yPos)` and radius `rad`

The `drawCircle` procedure is to be used to draw the following figure on a coordinate grid.



Which of the following code segments can be used to draw the figure?

Select <u>two</u> answers.

(A)
```
x ← 4
y ← 1
r ← 0
REPEAT 3 TIMES
{
   drawCircle(x, y, r)
   r ← r + 1
   y ← y + 1
}
```

(B)
```
x ← 4
y ← 1
r ← 0
REPEAT 3 TIMES
{
   r ← r + 1
   y ← y + 1
   drawCircle(x, y, r)
}
```

(C)
```
x ← 4
y ← 4
r ← 3
REPEAT 3 TIMES
{
   drawCircle(x, y, r)
   y ← y - 1
   r ← r - 1
}
```

(D)
```
x ← 4
y ← 4
r ← 3
REPEAT 3 TIMES
{
   y ← y - 1
   r ← r - 1
   drawCircle(x, y, r)
}
```


© 2023 College Board

18. In the following statement, `val1`, `val2`, and `result` are Boolean variables.

```
result ← val1 AND NOT val2
```

Which of the following code segments produce the same result as the statement above for all possible values of `val1` and `val2`?

Select <u>two</u> answers.

(A)

```
result ← false
IF val1
    IF NOT val2
        result ← true
```

(B)

```
result ← false
IF val1
    result ← true
IF NOT val2
    result ← true
```

(C)

```
IF val1
    result ← true
ELSE
    IF val2
        result ← false
    ELSE
        result ← true
```

(D)

```
IF val1
    IF val2
        result ← false
    ELSE
        result ← true
ELSE
    result ← false
```


© 2023 College Board

Section II: Create Performance Task: Written Responses

The Create performance task section of the end-of-course exam consists of two questions that require students to write responses that demonstrate understanding of their personal Create performance task. The two written response questions include four distinct prompts: Written Response 1, Written Response 2(a), Written Response 2(b), and Written Response 2(c). The following are sample prompts – but the specific prompts will vary across the different versions of the exam. Students will have access to their student-authored Personalized Project Reference, as long as it was submitted as final via the AP Digital Portfolio, when responding to these prompts.

Prompt Category	Sample Prompt
Written Response 1: Program Design, Function, and Purpose	Identify an expected user of your program. Describe one way your program's design meets the needs of this user.
Written Response 2(a): Algorithm Development	Consider the first iteration statement included in the Procedure section of your Personalized Project Reference. Identify the number of times the body of your iteration statement will execute. Describe a condition or error that would cause your iteration statement to not terminate and cause an infinite loop. If no such condition or error exists, explain how the loop could be modified to cause an infinite loop.
Written Response 2(b): Errors and Testing	Consider the procedure included in part (i) of the Procedure section of your Personalized Project Reference. Describe a change to your procedure that will result in a run-time error. Explain why this change will result in a run-time error.
Written Response 2(c): Data and Procedural Abstraction	Suppose you are provided with a procedure called `isEqual(value1, value2)`. The procedure returns `true` if the two parameters `value1` and `value2` are equal in value and returns `false` otherwise. Using the list you identified in the List section of your Personalized Project Reference, explain in detailed steps an algorithm that uses `isEqual` to count the number of times a certain value appears in your list. Your explanation must be detailed enough for someone else to write the program code.

© 2023 College Board

Answer Key and Question Alignment to Course Framework

Multiple-Choice Question	Answer	Skill	Learning Objective
1	A	5.A	CSN-1.C
2	A	2.B	DAT-1.C.b
3	D	2.B	AAP-2.K.a
4	C	5.E	IOC-2.C
5	C	5.C	IOC-1.C
6	D	4.B	AAP-2.F.b
7	D	5.E	IOC-2.B
8	B	2.A	AAP-2.J
9	D	4.C	CRD-2.J
10	B	5.B	DAT-2.B
11	C	4.C	CRD-2.I.b
12	C	1.D	CSN-2.A.b
13	A	1.D	AAP-2.P.a
14	C	3.A	CRD-2.C
15	B	5.C	IOC-1.A
16	A	5.D	IOC-2.A
17	B,C	3.B	AAP-3.A.a
18	A,D	1.D	AAP-2.L

The scoring information for the questions within this course and exam description, along with further exam resources, can be found on the **AP Computer Science Principles Exam page** on AP Central.

© 2023 College Board

Scoring Guidelines

Video, Program Requirements, and Written Response 1 — 3 points

General Scoring Notes

- Responses should be evaluated solely on the rationale provided.
- Responses must demonstrate all criteria, including those within bulleted lists, in each row to earn the point for that row.
- Terms and phrases defined in the terminology list are italicized when they first appear in the scoring criteria.

© 2023 College Board

Video, Program Requirements, and Written Response 1 — 3 points

Reporting Category	Scoring Criteria	Decision Rules
Course Project: Video (0-1 points)	**The video demonstrates the running of the program including:** • *input* • *program functionality* • *output*	**Consider only the video when scoring this point.** **Do NOT award a point if the following is true:** • The video does not show a demonstration of the program running (screenshots or storyboards are not acceptable and would not be credited).
Course Project: Program Requirements (0-1 points)	**The program code includes:** • *A list* • *A procedure* • *A call to the procedure* • *Selection* • *Iteration*	**Consider the Personalized Project Reference (or Program Code if necessary) when scoring this point.** • If the program requirements do not appear in the Personalized Project Reference, consider the full program code file when scoring this point. • The procedure does not need to have a parameter to earn this point. • The code segments demonstrating selection and iteration do not need to appear in the same algorithm to earn this point. • The code segments demonstrating selection and iteration do not need to be contained in a procedure to earn this point. **Do NOT award a point if any one or more of the following is true:** • The list is a one-element list. • The use of the list is irrelevant or not used in the program. • The use of either the selection or the iteration is trivial (i.e., does not affect the outcome of the program).
Written Response 1: Program Design, Function, and Purpose (0-1 points)	**The written response:** • identifies an expected user of the program. • describes one way the program's design meets the needs of the identified user.	**Consider Written Response 1 and the student's Program Code when scoring this point.** • The response must relate the program design to the needs of the identified user. **Do NOT award a point if the following is true:** • The description of the design is implausible, inaccurate, or inconsistent with the program.

© 2023 College Board

Written Response 2 — 3 points

General Scoring Notes

- Responses should be evaluated solely on the rationale provided.
- Responses must demonstrate all criteria, including those within bulleted lists, in each row to earn the point for that row.
- Terms and phrases defined in the terminology list are italicized when they first appear in the scoring criteria.

© 2023 College Board

Written Response 2 — 3 points

Reporting Category	Scoring Criteria	Decision Rules
Written Response 2(a): Algorithm Development (0-1 points)	**The written response:** • identifies the number of times the body of the iteration statement will execute. AND • describes a condition or error that would cause an infinite loop. OR • if no such condition or error exists, explains how the loop could be modified to cause an infinite loop.	**Consider the Personalized Project Reference and Written Response 2(a) when scoring this point.** • If multiple iteration statements are included in the Procedure section of the Personalized Project Reference, use the first iteration statement to determine whether the point is earned. • The iteration statement does not need to be contained in a procedure to earn this point. **Do NOT award a point if the following is true:** • The identified number of times the body of the iteration statement will execute does not match the code.
Written Response 2(b): Errors and Testing (0-1 points)	**The written response:** • describes a change to the procedure that will result in a run-time error. • explains why the change will result in a run-time error.	**Consider the Personalized Project Reference and Written Response 2(b) when scoring this point.** • If multiple procedures are included in part (i) of the Procedure section of the Personalized Project Reference, use the first procedure to determine whether the point is earned. **Do NOT award a point if any one or more of the following is true:** • A procedure is not identified in part (i) of the Procedure section of the Personalized Project Reference. • The response does not apply to the procedure in part (i) of the Procedure section of the Personalized Project Reference. • The response describes expected behavior that is implausible, inaccurate, or inconsistent with the program.
Written Response 2(c): Data and Procedural Abstraction (0-1 points)	**The written response:** • explains in detailed steps an algorithm that uses `isEqual` to count the number of elements in the list that are equal to a certain value.	**Consider the Personalized Project Reference and Written Response 2(c) when scoring this point.** • If multiple lists are included in the List section of the Personalized Project Reference, use the first list to determine whether the point is earned. • The algorithm can be described in code, pseudocode, as a sequence of steps in English, or as a paragraph in English. • The algorithm must describe iterating over all elements of the list, calling `isEqual` on each list element and maintaining a count of the number of list elements equal to a certain value. • The algorithm must correctly determine the number of elements that are equal to a certain value. Some ways this can be determined is by storing the value in a variable, returning it, or displaying it to the user. **Do NOT award a point if the following is true:** • A list is not identified in the List section of the Personalized Project Reference. • The description of the algorithm is not detailed enough for someone else to write the code.

© 2023 College Board

AP Computer Science Principles Create Performance Task Terminology

Algorithm: An algorithm is a finite set of instructions that accomplish a specific task. Every algorithm can be constructed using combinations of sequencing, selection, and iteration.

Arguments: The values of the parameters when a procedure is called.

Collection type: Aggregates elements in a single structure. Some examples include: databases, hash tables, dictionaries, sets, or any other type that aggregates elements in a single structure.

Data stored in a list: Input into the list can be through an initialization or through some computation on other variables or list elements.

Input: Program input is data that are sent to a computer for processing by a program. Input can come in a variety of forms, such as tactile (through touch), audible, visual, or text. An event is associated with an action and supplies input data to a program.

Iteration: Iteration is a repetitive portion of an algorithm. Iteration repeats until a given condition is met or for a specified number of times. The use of recursion is a form of iteration.

List: A list is an ordered sequence of elements. The use of lists allows multiple related items to be represented using a single variable. Lists are referred to by different terms, such as arrays or arraylists, depending on the programming language.

List being used: Using a list means the program is creating new data from existing data or accessing multiple elements in the list.

Output: Program output is any data that are sent from a program to a device. Program output can come in a variety of forms, such as tactile, audible, visual, movement, or text.

Parameter: A parameter is an input variable of a procedure. Explicit parameters are defined in the procedure header. Implicit parameters are those that are assigned in anticipation of a call to the procedure. For example, an implicit parameter can be set through interaction with a graphical user interface.

Procedure: A procedure is a named group of programming instructions that may have parameters and return values. Procedures are referred to by different names, such as method, function, or constructor, depending on the programming language.

Program code segment: A code segment refers to a collection of program statements that are part of a program. For text-based, the collection of program statements should be continuous and within the same procedure. For block-based, the collection of program statements should be contained in the same starter block or what is referred to as a "Hat" block.

Program functionality: The behavior of a program during execution, often described by how a user interacts with it.

Purpose: The problem being solved or creative interest being pursued through the program.

Selection: Selection determines which parts of an algorithm are executed based on a condition being true or false. The use of try / exception statements is a form of selection statements.

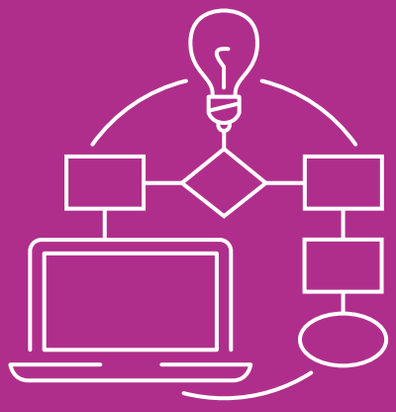
Sequencing: The application of each step of an algorithm in the order in which the code statements are given.

Student-developed procedure / algorithm: Program code that is student developed has been written (individually or collaboratively) by the student who submitted the response. Calls to existing program code or libraries can be included but are not considered student developed. Event handlers are built-in abstractions in some languages and will therefore not be considered student-developed. In some block-based programming languages, event handlers begin with "when".

© 2023 College Board

AP COMPUTER SCIENCE PRINCIPLES

Student Handouts



AP COMPUTER SCIENCE PRINCIPLES

Student Handouts

The following pages contain a student-directed version of the performance task guidelines that you can print out or copy to share with your students.

© 2023 College Board

THIS PAGE IS INTENTIONALLY LEFT BLANK.

Create Performance Task

Programming is a collaborative and creative process that brings ideas to life through the development of software. In the Create performance task, you will design and implement a program that might solve a problem, enable innovation, explore personal interests, or express creativity. Your submission must include the elements listed in the Submission Requirements section below.

You are allowed to collaborate with your partner(s) on the development of the program only. **The video and Personalized Project Reference that you submit for this performance task must be completed individually, without any collaboration with your partner(s) or anyone else.** You can develop the code segments used in your **Personalized Project Reference** with your partner(s) or on your own as you work on the performance task during class.

Please note that once your teacher has assigned this performance task as one of your AP score components, you are expected to complete the task without assistance from anyone except for your partner(s) and then only when developing the program code. You must follow the Guidelines for Completing the Create Performance Task section below.

General Requirements

You will be provided with a minimum of 9 hours of class time to complete and submit the following:

- **Final program code** (created independently or collaboratively)
- **A video that displays the running of your program and demonstrates functionality you developed** (created independently)
- **Code Segments for your Personalized Project Reference** (created independently)

Note: Students in nontraditional classroom environments should consult a school-based AP Coordinator for instructions.

© 2023 College Board

Submission Requirements

COMPONENT A: PROGRAM CODE (CREATED INDEPENDENTLY OR COLLABORATIVELY)

Submit one PDF file that contains all of your program code (including comments). Include comments or acknowledgments for any part of the submitted program code that has been written by someone other than you and/or your collaborative partner(s).

IMPORTANT:

If the programming environment allows you to include comments, this is the preferred way to acknowledge and give credit to another author. However, if the programming environment does not allow you to include comments, you can add them in a document editor when you capture your program code for submission.

In your program, you must include student-developed program code that contains the following:

- ☐ Instructions for input from one of the following:
  - the user (including user actions that trigger events)
  - a device
  - an online data stream
  - a file
- ☐ Use of at least one **list** (or other **collection type**) to represent a collection of data that is stored and used to manage program complexity and help fulfill the program's purpose

  IMPORTANT:

  The data abstraction must make the program easier to develop (alternatives would be more complex) or easier to maintain (future changes to the size of the list would otherwise require significant modifications to the program code).
- ☐ At least one procedure that contributes to the program's intended purpose, where you have defined:
  - the procedure's name
  - the return type (if necessary)
  - one or more parameters

  IMPORTANT:

  Implementation of built-in or existing procedures or language structures, such as event handlers or main methods, are not considered student-developed.
- ☐ An algorithm that includes sequencing, selection, and iteration that is in the body of the selected procedure
- ☐ Calls to your student-developed procedure
- ☐ Instructions for output (tactile, audible, visual, or textual) based on input and program functionality

DEFINITION:

List

A **list** is an ordered sequence of elements. The use of lists allows multiple related items to be represented using a single variable. Lists may be referred to by different names, such as **arrays**, depending on the programming language.

DEFINITION:

Collection Type

A **collection type** is a type that aggregates elements in a single structure. Some examples include lists, databases, and sets.

IMPORTANT:

With text-based program code, you can use the print command to save your program code as a PDF file, or you can copy and paste your code to a text document and then convert it into a PDF file. With block-based program code, you can create screen captures that include only your program code, paste these images into a document, and then convert that document to a PDF. Screen captures should not be blurry, and text should be at least 10 pt font size.

© 2023 College Board

COMPONENT B: VIDEO (CREATED INDEPENDENTLY) Submit one video file that demonstrates the running of your program as described below. Collaboration is not allowed during the development of your video.

Your video must demonstrate your program running, including:

- ☐ Input to your program
- ☐ At least one aspect of the functionality of your program
- ☐ Output produced by your program

Your video may NOT contain:

- ☐ Any distinguishing information about yourself
- ☐ Voice narration (though text captions are allowed)

Your video must be:

- ☐ Either .webm, .mp4, .wmv, .avi, or .mov format
- ☐ No more than 1 minute in length
- ☐ No more than 30MB in file size

© 2023 College Board

COMPONENT C: PERSONALIZED PROJECT REFERENCE (CREATED INDEPENDENTLY) To assist in responding to the written response prompts on exam day, submit required portions of your code by capturing and pasting program code segments you developed during the administration of this task. Screen captures should not be blurry, and text should be at least 10-point font size. Your code segments should not include any comments. These code segments will be made available to you on exam day only if this component is submitted as final in the AP Digital Portfolio by the deadline.

Procedure: Capture and paste two program code segments you developed during the administration of this task that contain a student-developed procedure that implements an algorithm used in your program and a call to that procedure.

IMPORTANT:

Built-in or existing procedures and language structures, such as event handlers and main methods, are not considered student-developed.

i. The first program code segment must be a student-developed procedure that:

- ☐ Defines the procedure's name and return type (if necessary)
- ☐ Contains and uses one or more parameters that have an effect on the functionality of the procedure
- ☐ Implements an algorithm that includes sequencing, selection, and iteration

ii. The second program code segment must show where your student-developed procedure is being called in your program.

© 2023 College Board

List: Capture and paste two program code segments you developed during the administration of this task that contain a list (or other collection type) being used to manage complexity in your program.

i. The first program code segment must show how data have been stored in the list.

ii. The second program code segment must show the data in the same list being used, such as creating new data from the existing data or accessing multiple elements in the list, as part of fulfilling the program's purpose.

DEFINITION:

List

A **list** is an ordered sequence of elements. The use of lists allows multiple related items to be represented using a single variable. Lists may be referred to by different names, such as **arrays**, depending on the programming language.

DEFINITION:

Collection Type

A **collection type** is a type that aggregates elements in a single structure. Some examples include lists, databases, and sets.

IMPORTANT:

The data abstraction manages complexity by making the program easier to develop (alternatives would be more complex) or easier to maintain (future changes to the size of the list would otherwise require significant modifications to the program code).

© 2023 College Board

THIS PAGE IS INTENTIONALLY LEFT BLANK.

Guidelines for Completing the Performance Task

Academic Integrity and Plagiarism Policy

This policy addresses plagiarism and academic integrity in completing the Create Performance Task.

Plagiarism

The use of program code, media (e.g., video, images, sound), data, information, or evidence created by someone else or with generative AI tools in the creation of a program and/or a program code segment(s), without appropriate acknowledgment (i.e., through citation, through attribution, and/or by reference), is considered **plagiarism**. A student who commits plagiarism will receive a score of 0 on the Create performance task, including their responses to the written response prompts on the end-of-course AP Exam.

To the best of their ability, teachers will ensure that students understand how to ethically incorporate ideas that are not their own and provide credit to the original creator or source, as well as the consequences of plagiarism.

Acceptable Generative AI Use

Students are permitted to utilize generative AI tools as supplementary resources for understanding coding principles, assisting in code development, and debugging. This responsible use aligns with current guidelines for peer collaboration on developing code.

Students should be aware that generative AI tools can produce incomplete code, code that creates or introduces biases, code with errors, inefficiencies in how the code executes, or code complexities that make it difficult to understand and therefore explain the code. It is the student's responsibility to review and understand any code co-written with AI tools, ensuring its functionality. Additionally, students must be prepared to explain their code in detail, as required on the end-of-course AP Exam.

Preparing for Final Submission

Students are not permitted to collaborate on the video or creation of the Personalized Project Reference.

The Personalized Project Reference cannot include course content or comments within the code or on any other part of the reference. Including course content or comments in the Personalized Project Reference will result in students receiving a score of 0 on the Create performance task, including their responses to the written response prompts on the end-of-course AP Exam.

© 2023 College Board

Attestations

During the final submission process in the AP Digital Portfolio, students will be asked to attest that they have followed the Performance Task guidelines and have not plagiarized their submission. Each of the three components of the Create performance task must be submitted as final to be sent for scoring. Additionally, if students do not submit their Personalized Project Reference by the deadline, they will not have this resource available on exam day to complete their written response section.

© 2023 College Board

Preparing for the Performance Task

*Prior to beginning the performance task, you **should**:*

- Obtain content knowledge and skills that will help you succeed on the performance task. This can include, but needs not be limited to, the iterative development process, strategies for collaboration, the development of both data and procedural abstractions, describing an algorithm's purpose and explaining how it functions, and identifying test data that demonstrates the different outcomes of an algorithm. A development process includes exploration, investigation, reflection, design, implementation, and testing your program.
- Review the performance task directions and guidelines.
- Brainstorm problems that programming can address, or brainstorm special interests that programming can help develop.
- As needed, seek assistance from your teacher or AP Coordinator on defining your focus and choice of topics.
- Be prepared to collaborate with peers as necessary.
- Practice and discuss all or part of the performance task, including potential written response prompts that could appear on the end-of-course AP Exam.
- Review the role your teacher can and cannot play in providing assistance during the actual performance task, and take advantage of the opportunity to get assistance and feedback from your teacher during practice.
- Ensure you know the proper way to cite media or program code, including APIs or other pieces of open-source code, used in the creation of your program. Any media or program code that has not been written by you must be cited, and credit must be given to the author. Any existing program code needs to be extended in some new way by adding new functionality.
- Examples of responses can be found on the **AP Computer Science Principles Exam page**. You cannot submit any work from AP Central samples, curriculum provider samples, or practice performance tasks for your final submission.
- Be aware that the scoring process that occurs in the AP Reading may be different from the scoring process that occurs in your classroom; the AP score that you receive may be different than your classroom grade.
- Read through the **AP Digital Portfolio file submission requirements** and process, paying attention to the instructions concerning the file type, size, and length to be uploaded. This is important to ensure that work is sent for AP scoring. This process includes:
  - uploading your files to the correct part of the task
  - submitting each component as final
  - completing the College Board attestations to the originality of your work

  All three components must be submitted as final by the deadline posted on AP Central. Only files that are submitted as final will be sent for scoring. Additionally, if you do not submit your Personalized Project Reference by the deadline, you will not have this resource available on exam day to complete your written response section.

© 2023 College Board

- Practice creating a video of your program running, while adhering to file type, size, and length requirements.
- Practice making readable screen captures and creating a PDF file of your program code to submit for the performance task:
  - With text-based program code, you can use the print command to save your program code as a PDF file, or you can copy and paste your code to a text document and then convert it into a PDF file.
  - With block-based program code, you can create screen captures that include only your program code, paste these images into a document, and then convert that document to a PDF file. Screen captures should not be blurry, and text should be at least 10 pt font size.
  - For the Personalized Project Reference, screen captures should not be blurry, and text should be at least 10 pt font size. Any course content or comments that were included with the program code, should not be included in the screen captures used in the Personalized Project Reference. Including course content or comments in the Personalized Project Reference could result in students receiving a 0 for the Create performance task.
- Understand that you may not revise your work once you have submitted it as final to the AP Digital Portfolio.
- Review the categories for each written response prompts on the end-of-course AP Exam. The written response prompts will be related to your Create performance task. Practice writing responses for these types of questions.

© 2023 College Board

Completing the Performance Task

*You **must:***

- Submit all three components of your performance task as final prior to the submission deadline, which can be found on the AP Computer Science Principles Exam page on AP Central. Only files submitted as final will be sent for scoring. If you do not submit your Personalized Project Reference by the deadline, you will not have this resource available on exam day to complete your written response section.
- Follow a calendar or schedule that provides time for all performance task components to be completed and uploaded in advance of the deadline.
- Read the performance task directions.
- Apply the computer science knowledge you have obtained throughout the course, and when completing the performance task.
- Use acceptable acknowledgment practices when using program code, media (i.e., images, videos, sound), or data sources created by others or with generative AI tools in your program code to avoid plagiarism. Program code not written by you could include starter code provided by your teacher, the use of APIs or open-source code, or generated using an AI tool. When using existing code, you must extend the program in some new way by adding new functionality. Some examples of ways to provide attribution for program code that was not authored by you are as follows:
  - If the program code has been made available for your use by your teacher, add a comment that states: *This code was provided as starter code by my teacher*.
  - If the program code has been made available through an API or open-source code, add a comment that states: *This code was made freely available by [source of code]*.
  - If the program code has been co-created with the assistance of a generative AI tool, add a comment that states: *This code was generated using [Generative AI Tool Name]*.
- Add comments to your program code to clarify the functionality of program code segments or to acknowledge and credit authors of media, data sources, or program code:
  - If the programming environment allows you to include comments, this is the preferred way to acknowledge and give credit to another author.
  - If the programming environment does not allow you to include comments, you can add them in a document editor when you capture your program code for submission.

© 2023 College Board

*Once you have started your official administration of the performance task, you **may not**:*

- Seek assistance in writing, revising, amending, or correcting your work, including debugging the program, writing or designing functionality in the program, testing the program, or making revisions to the program, from anyone other than your collaborative partner(s).
- Submit practice performance tasks or any work that has been revised, amended, or corrected by another individual, other than your collaborative partner(s) or cited program code, as a submission for AP Exam scoring.
- Include any course content in the screen captures of the program code included in the Personalized Project Reference. Any course content or comments that were included in the program code during the development of your program should be removed prior to taking screen captures. Including course content or comments in your screen captures will result in the written response portion of your exam being scored a 0.
- Collaborate during the creation of your video or creation of your Personalized Project Reference.
- Revise your work once you have completed and submitted it as final to the AP Digital Portfolio.

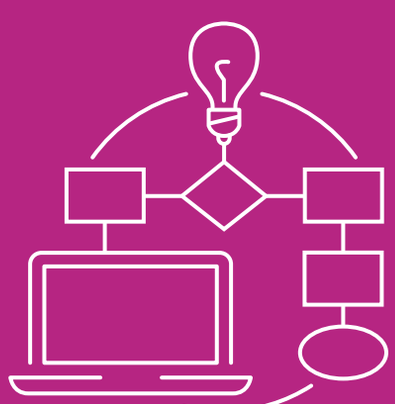
*Once you have started your official administration of the performance task, you **may**:*

- Collaborate with your partner(s) only during the ideation and development, including debugging and error testing, of your program code, if you choose to do so. NOTE: You are not allowed to collaborate on your video or creation of your Personalized Project Reference.
- Follow a timeline and schedule for completing the performance task.
- Seek assistance from your teacher or AP Coordinator on the formation of groups and resolution of collaboration issues when one collaborative partner is clearly and directly impeding the completion of the performance task.
- Seek clarification from your teacher or AP Coordinator on the program requirements and submission requirements for the performance task when you do not understand the directions.
- Work on the performance task outside of designated class time.
- Seek assistance from your teacher or AP Coordinator to resolve technical problems that impede work, such as a failing workstation or difficulty with access to networks, or to help with saving files or making movie files.
- Keep a programming journal of the design choices that were made during the development of the program code or code segment and the effect of these decisions on the program's function.

© 2023 College Board

AP COMPUTER SCIENCE PRINCIPLES

Appendix



AP COMPUTER SCIENCE PRINCIPLES

Appendix 1: AP CSP Exam Reference Sheet

© 2023 College Board

THIS PAGE IS INTENTIONALLY LEFT BLANK.

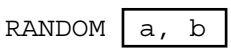
Exam Reference Sheet

Instruction	Explanation
Assignment, Display, and Input	
Text: `a ← expression` Block: 	Evaluates `expression` and then assigns a copy of the result to the variable `a`.
Text: `DISPLAY(expression)` Block: 	Displays the value of `expression`, followed by a space.

Text:
`INPUT()`
Block:
`INPUT`
	Accepts a value from the user and returns the input value.

Arithmetic Operators and Numeric Procedures	

Text and Block:
`a + b`
`a - b`
`a * b`
`a / b`
	The arithmetic operators `+`, `-`, `*`, and `/` are used to perform arithmetic on `a` and `b`.
	For example, `17 / 5` evaluates to `3.4`.
	The order of operations used in mathematics applies when evaluating expressions.

Text and Block: `a MOD b`	Evaluates to the remainder when `a` is divided by `b`. Assume that `a` is an integer greater than or equal to `0` and `b` is an integer greater than `0`. For example, `17 MOD 5` evaluates to `2`. The `MOD` operator has the same precedence as the `*` and `/` operators.
Text: `RANDOM(a, b)` Block: 	Generates and returns a random integer from `a` to `b`, including `a` and `b`. Each result is equally likely to occur. For example, `RANDOM(1, 3)` could return `1`, `2`, or `3`.
Relational and Boolean Operators	

Text and Block:
`a = b`
`a ≠ b`
`a > b`
`a < b`
`a ≥ b`
`a ≤ b`
	The relational operators `=`, `≠`, `>`, `<`, `≥`, and `≤` are used to test the relationship between two variables, expressions, or values. A comparison using relational operators evaluates to a Boolean value.
	For example, `a = b` evaluates to `true` if `a` and `b` are equal; otherwise it evaluates to `false`.

© 2023 College Board

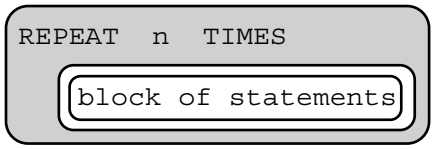
Instruction	Explanation
Relational and Boolean Operators (continued)	
Text: `NOT condition` Block: NOT (condition)	Evaluates to `true` if `condition` is `false`; otherwise evaluates to `false`.
Text: `condition1 AND condition2` Block: (condition1) AND (condition2)	Evaluates to `true` if both `condition1` and `condition2` are `true`; otherwise evaluates to `false`.
Text: `condition1 OR condition2` Block: (condition1) OR (condition2)	Evaluates to `true` if `condition1` is `true` or if `condition2` is `true` or if both `condition1` and `condition2` are `true`; otherwise evaluates to `false`.
Selection	

Text:
`IF(condition)`
`{`
`  <block of statements>`
`}`
Block:
IF (condition)
(block of statements)
	The code in `block of statements` is executed if the Boolean expression `condition` evaluates to `true`; no action is taken if `condition` evaluates to `false`.

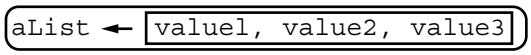
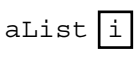
Text:
`IF(condition)`
`{`
`  <first block of statements>`
`}`
`ELSE`
`{`
`  <second block of statements>`
`}`
Block:
IF (condition)
(first block of statements)
ELSE
(second block of statements)
	The code in `first block of statements` is executed if the Boolean expression `condition` evaluates to `true`; otherwise the code in `second block of statements` is executed.

© 2023 College Board

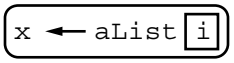
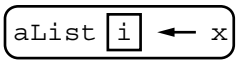
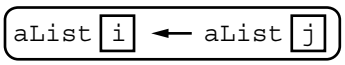
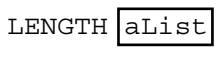
Instruction	Explanation
Iteration	

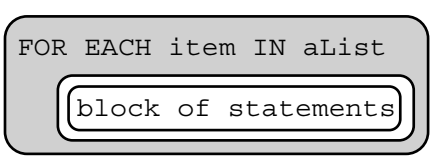
Text:
`REPEAT n TIMES`
`{`
`  <block of statements>`
`}`
Block:
 
	The code in `block of statements` is executed `n` times.

Text:
`REPEAT UNTIL(condition)`
`{`
`  <block of statements>`
`}`
Block:
 	
	The code in `block of statements` is repeated until the Boolean expression `condition` evaluates to `true`.

List Operations	
For all list operations, if a list index is less than 1 or greater than the length of the list, an error message is produced and the program terminates.	
Text: `aList ← [value1, value2, value3, ...]` Block: 	Creates a new list that contains the values `value1`, `value2`, `value3`, and `...` at indices `1`, `2`, `3`, and `...` respectively and assigns it to `aList`.
Text: `aList ← []` Block: 	Creates an empty list and assigns it to `aList`.
Text: `aList ← bList` Block: 	Assigns a copy of the list `bList` to the list `aList`. For example, if `bList` contains `[20, 40, 60]`, then `aList` will also contain `[20, 40, 60]` after the assignment.
Text: `aList[i]` Block: 	Accesses the element of `aList` at index `i`. The first element of `aList` is at index `1` and is accessed using the notation `aList[1]`.

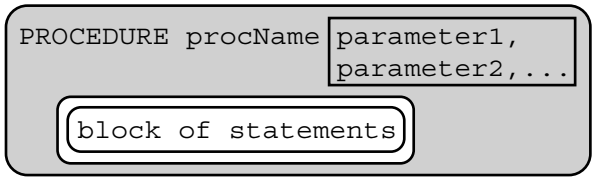
© 2023 College Board

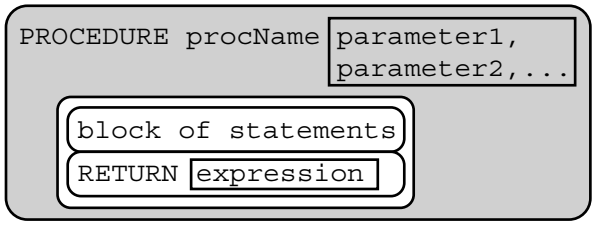
Instruction	Explanation
List Operations (continued)	
Text: `x ← aList[i]` Block: 	Assigns the value of `aList[i]` to the variable `x`.
Text: `aList[i] ← x` Block: 	Assigns the value of `x` to `aList[i]`.
Text: `aList[i] ← aList[j]` Block: 	Assigns the value of `aList[j]` to `aList[i]`.
Text: `INSERT(aList, i, value)` Block: 	Any values in `aList` at indices greater than or equal to `i` are shifted one position to the right. The length of the list is increased by 1, and `value` is placed at index `i` in `aList`.
Text: `APPEND(aList, value)` Block: 	The length of `aList` is increased by 1, and `value` is placed at the end of `aList`.
Text: `REMOVE(aList, i)` Block: 	Removes the item at index `i` in `aList` and shifts to the left any values at indices greater than `i`. The length of `aList` is decreased by 1.
Text: `LENGTH(aList)` Block: 	Evaluates to the number of elements in `aList`.

Text:
`FOR EACH item IN aList`
`{`
`  <block of statements>`
`}`
Block:
 
	The variable `item` is assigned the value of each element of `aList` sequentially, in order, from the first element to the last element. The code in `block of statements` is executed once for each assignment of `item`.

© 2023 College Board

Instruction	Explanation
Procedures and Procedure Calls	

Text:
`PROCEDURE procName(parameter1, parameter2, ...)`
`{`
`<block of statements>`
`}`
Block:
 
	Defines `procName` as a procedure that takes zero or more arguments. The procedure contains `block of statements`.
	The procedure `procName` can be called using the following notation, where `arg1` is assigned to `parameter1`, `arg2` is assigned to `parameter2`, etc.:
	`procName(arg1, arg2, ...)`

Text:
`PROCEDURE procName(parameter1, parameter2, ...)`
`{`
`<block of statements>`
`RETURN(expression)`
`}`
Block:
 
	Defines `procName` as a procedure that takes zero or more arguments. The procedure contains `block of statements` and returns the value of `expression`. The `RETURN` statement may appear at any point inside the procedure and causes an immediate return from the procedure back to the calling statement.
	The value returned by the procedure `procName` can be assigned to the variable `result` using the following notation:
	`result ← procName(arg1, arg2, ...)`

Text: `RETURN(expression)` Block: 	Returns the flow of control to the point where the procedure was called and returns the value of `expression`.
Robot	
If the robot attempts to move to a square that is not open or is beyond the edge of the grid, the robot will stay in its current location and the program will terminate.	
Text: `MOVE_FORWARD()` Block: 	The robot moves one square forward in the direction it is facing.
Text: `ROTATE_LEFT()` Block: 	The robot rotates in place 90 degrees counterclockwise (i.e., makes an in-place left turn).

© 2023 College Board

Instruction	Explanation
Robot	

Text:
`ROTATE_RIGHT()`
Block:
`ROTATE_RIGHT`
	The robot rotates in place 90 degrees clockwise (i.e., makes an in-place right turn).

Text:
`CAN_MOVE(direction)`
Block:
`CAN_MOVE` `direction`
	Evaluates to `true` if there is an open square one square in the direction relative to where the robot is facing; otherwise evaluates to `false`. The value of `direction` can be `left`, `right`, `forward`, or `backward`.

© 2023 College Board

AP COMPUTER SCIENCE PRINCIPLES

Appendix 2: AP CSP Conceptual Framework

© 2023 College Board

THIS PAGE IS INTENTIONALLY LEFT BLANK.

Conceptual Framework

Big Idea 1: Creative Development (CRD)

When developing computing innovations, developers can use a formal, iterative design process or experimentation. While using either approach, developers will encounter phases of investigating and reflecting, designing, prototyping, and testing. Additionally, collaboration is an important tool to use at any phase of development because considering multiple perspectives allows for improvement of innovations.

Enduring Understanding	Learning Objective	Essential Knowledge
CRD-1 *Incorporating multiple perspectives through collaboration improves computing innovations as they are developed.*	**CRD-1.A** Explain how computing innovations are improved through collaboration. 1.C	**CRD-1.A.1** A computing innovation includes a program as an integral part of its function.
		CRD-1.A.2 A computing innovation can be physical (e.g., self-driving car), nonphysical computing software (e.g., picture editing software), or a nonphysical computing concept (e.g., e-commerce).
		CRD-1.A.3 Effective collaboration produces a computing innovation that reflects the diversity of talents and perspectives of those who designed it.
		CRD-1.A.4 Collaboration that includes diverse perspectives helps avoid bias in the development of computing innovations.
		CRD-1.A.5 Consultation and communication with users are important aspects of the development of computing innovations.
		CRD-1.A.6 Information gathered from potential users can be used to understand the purpose of a program from diverse perspectives and to develop a program that fully incorporates these perspectives.
	CRD-1.B Explain how computing innovations are developed by groups of people. 1.C	**CRD-1.B.1** Online tools support collaboration by allowing programmers to share and provide feedback on ideas and documents.
		CRD-1.B.2 Common models such as pair programming exist to facilitate collaboration.
	CRD-1.C Demonstrate effective interpersonal skills during collaboration. 1.C	**CRD-1.C.1** Effective collaborative teams practice interpersonal skills, including but not limited to: ▪ communication ▪ consensus building ▪ conflict resolution ▪ negotiation

continued on next page

© 2023 College Board

Big Idea 1: Creative Development (CRD) *(cont'd)*

Enduring Understanding	Learning Objective	Essential Knowledge
CRD-2 *Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/ feedback cycles, and that leaves ample room for experimentation and risk-taking.*	**CRD-2.A** Describe the purpose of a computing innovation. 1.A	**CRD-2.A.1** The purpose of computing innovations is to solve problems or to pursue interests through creative expression.
		CRD-2.A.2 An understanding of the purpose of a computing innovation provides developers with an improved ability to develop that computing innovation.
	CRD-2.B Explain how a program or code segment functions. 4.A	**CRD-2.B.1** A *program* is a collection of program statements that performs a specific task when run by a computer. A program is often referred to as *software*.
		CRD-2.B.2 A *code segment* is a collection of program statements that is part of a program.
		CRD-2.B.3 A program needs to work for a variety of inputs and situations.
		CRD-2.B.4 The *behavior* of a program is how a program functions during execution and is often described by how a user interacts with it.
		CRD-2.B.5 A program can be described broadly by what it does, or in more detail by both what the program does and how the program statements accomplish this function.
	CRD-2.C Identify input(s) to a program. 3.A	**CRD-2.C.1** *Program inputs* are data sent to a computer for processing by a program. Input can come in a variety of forms, such as tactile, audio, visual, or text.
		CRD-2.C.2 An *event* is associated with an action and supplies input data to a program.
		CRD-2.C.3 Events can be generated when a key is pressed, a mouse is clicked, a program is started, or any other defined action occurs that affects the flow of execution.
		CRD-2.C.4 Inputs usually affect the output produced by a program.
		CRD-2.C.5 In event-driven programming, program statements are executed when triggered rather than through the sequential flow of control.
		CRD-2.C.6 Input can come from a user or other programs.

continued on next page

© 2023 College Board

Enduring Understanding	Learning Objective	Essential Knowledge
CRD-2 *Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/ feedback cycles, and that leaves ample room for experimentation and risk-taking.*	**CRD-2.D** Identify output(s) produced by a program. 3.A	**CRD-2.D.1** *Program outputs* are any data sent from a program to a device. Program output can come in a variety of forms, such as tactile, audio, visual, or text. **CRD-2.D.2** Program output is usually based on a program's input or prior state (e.g., internal values).
	CRD-2.E Develop a program using a development process. 1.B	**CRD-2.E.1** A development process can be ordered and intentional, or exploratory in nature. **CRD-2.E.2** There are multiple development processes. The following phases are commonly used when developing a program: ▪ investigating and reflecting ▪ designing ▪ prototyping ▪ testing **CRD-2.E.3** A development process that is iterative requires refinement and revision based on feedback, testing, or reflection throughout the process. This may require revisiting earlier phases of the process. **CRD-2.E.4** A development process that is incremental is one that breaks the problem into smaller pieces and makes sure each piece works before adding it to the whole.
	CRD-2.F Design a program and its user interface. 1.B	**CRD-2.F.1** The design of a program incorporates investigation to determine its requirements. **CRD-2.F.2** Investigation in a development process is useful for understanding and identifying the program constraints, as well as the concerns and interests of the people who will use the program. **CRD-2.F.3** Some ways investigation can be performed are as follows: ▪ collecting data through surveys ▪ user testing ▪ interviews ▪ direct observations

continued on next page

© 2023 College Board

Big Idea 1: Creative Development (CRD) *(cont'd)*

Enduring Understanding	Learning Objective	Essential Knowledge
CRD-2 *Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/ feedback cycles, and that leaves ample room for experimentation and risk-taking.*		**CRD-2.F.4** Program requirements describe how a program functions and may include a description of user interactions that a program must provide.
		CRD-2.F.5 A program's specification defines the requirements for the program.
		CRD-2.F.6 In a development process, the design phase outlines how to accomplish a given program specification.
		CRD-2.F.7 The design phase of a program may include: ▪ brainstorming ▪ planning and storyboarding ▪ organizing the program into modules and functional components ▪ creation of diagrams that represent the layouts of the user interface ▪ development of a testing strategy for the program
	CRD-2.G Describe the purpose of a code segment or program by writing documentation. **4.A**	**CRD-2.G.1** *Program documentation* is a written description of the function of a code segment, event, procedure, or program and how it was developed.
		CRD-2.G.2 *Comments* are a form of program documentation written into the program to be read by people and do not affect how a program runs.
		CRD-2.G.3 Programmers should document a program throughout its development.
		CRD-2.G.4 Program documentation helps in developing and maintaining correct programs when working individually or in collaborative programming environments.
		CRD-2.G.5 Not all programming environments support comments, so other methods of documentation may be required.

continued on next page

© 2023 College Board

Big Idea 1: Creative Development (CRD) *(cont'd)*

Enduring Understanding	Learning Objective	Essential Knowledge
CRD-2 *Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/ feedback cycles, and that leaves ample room for experimentation and risk-taking.*	**CRD-2.H** Acknowledge code segments used from other sources. 1.C	**CRD-2.H.1** It is important to acknowledge any code segments that were developed collaboratively or by another source. **CRD-2.H.2** Acknowledgement of a code segment(s) written by someone else and used in a program can be in the program documentation. The acknowledgement should include the origin or original author's name.
	CRD-2.I For errors in an algorithm or program: a. Identify the error. 4.C b. Correct the error. 4.C	**CRD-2.I.1** A *logic error* is a mistake in the algorithm or program that causes it to behave incorrectly or unexpectedly. **CRD-2.I.2** A *syntax error* is a mistake in the program where the rules of the programming language are not followed. **CRD-2.I.3** A *run-time error* is a mistake in the program that occurs during the execution of a program. Programming languages define their own run-time errors. **CRD-2.I.4** An *overflow error* is an error that occurs when a computer attempts to handle a number that is outside of the defined range of values. **CRD-2.I.5** The following are effective ways to find and correct errors: ▪ test cases ▪ hand tracing ▪ visualizations ▪ debuggers ▪ adding extra output statement(s)
	CRD-2.J Identify inputs and corresponding expected outputs or behaviors that can be used to check the correctness of an algorithm or program. 4.C	**CRD-2.J.1** In the development process, *testing* uses defined inputs to ensure that an algorithm or program is producing the expected outcomes. Programmers use the results from testing to revise their algorithms or programs. **CRD-2.J.2** Defined inputs used to test a program should demonstrate the different expected outcomes that are at or just beyond the extremes (minimum and maximum) of input data. **CRD-2.J.3** Program requirements are needed to identify appropriate defined inputs for testing.

© 2023 College Board

Big Idea 2: Data (DAT)

Data are central to computing innovations because they communicate initial conditions to programs and represent new knowledge. Computers consume data, transform data, and produce new data, allowing users to create new information or knowledge to solve problems through the interpretation of these data. Computers store data digitally, which means that the data must be manipulated in order to be presented in a useful way to the user.

Enduring Understanding	Learning Objective	Essential Knowledge
DAT-1 *The way a computer represents data internally is different from the way the data are interpreted and displayed for the user. Programs are used to translate data into a representation more easily understood by people.*	**DAT-1.A** Explain how data can be represented using bits. 3.C	**DAT-1.A.1** Data values can be stored in variables, lists of items, or standalone constants and can be passed as input to (or output from) procedures. **DAT-1.A.2** Computing devices represent data digitally, meaning that the lowest-level components of any value are bits. **DAT-1.A.3** *Bit* is shorthand for *binary digit* and is either 0 or 1. **DAT-1.A.4** A *byte* is 8 bits. **DAT-1.A.5** *Abstraction* is the process of reducing complexity by focusing on the main idea. By hiding details irrelevant to the question at hand and bringing together related and useful details, abstraction reduces complexity and allows one to focus on the idea. **DAT-1.A.6** Bits are grouped to represent abstractions. These abstractions include, but are not limited to, numbers, characters, and color. **DAT-1.A.7** The same sequence of bits may represent different types of data in different contexts. **DAT-1.A.8** Analog data have values that change smoothly, rather than in discrete intervals, over time. Some examples of analog data include pitch and volume of music, colors of a painting, or position of a sprinter during a race. **DAT-1.A.9** The use of digital data to approximate real-world analog data is an example of abstraction. **DAT-1.A.10** Analog data can be closely approximated digitally using a *sampling technique*, which means measuring values of the analog signal at regular intervals called *samples*. The samples are measured to figure out the exact bits required to store each sample.

continued on next page

© 2023 College Board

Big Idea 2: Data (DAT) *(cont'd)*

Enduring Understanding	Learning Objective	Essential Knowledge
DAT-1 *The way a computer represents data internally is different from the way the data are interpreted and displayed for the user. Programs are used to translate data into a representation more easily understood by people.*	**DAT-1.B** Explain the consequences of using bits to represent data. 1.D	**DAT-1.B.1** In many programming languages, integers are represented by a fixed number of bits, which limits the range of integer values and mathematical operations on those values. This limitation can result in overflow or other errors. **DAT-1.B.2** Other programming languages provide an abstraction through which the size of representable integers is limited only by the size of the computer's memory; this is the case for the language defined in the exam reference sheet. **DAT-1.B.3** In programming languages, the fixed number of bits used to represent real numbers limits the range and mathematical operations on these values; this limitation can result in round-off and other errors. Some real numbers are represented as approximations in computer storage. **EXCLUSION STATEMENT (EK DAT-1.B.3)** *Specific range limitations for real numbers are outside the scope of this course and the AP Exam.*
	DAT-1.C For binary numbers: a. Calculate the binary (base 2) equivalent of a positive integer (base 10) and vice versa. 2.B b. Compare and order binary numbers. 2.B	**DAT-1.C.1** Number bases, including binary and decimal, are used to represent data. **DAT-1.C.2** Binary (base 2) uses only combinations of the digits zero and one. **DAT-1.C.3** Decimal (base 10) uses only combinations of the digits 0 – 9. **DAT-1.C.4** As with decimal, a digit's position in the binary sequence determines its numeric value. The numeric value is equal to the bit's value (0 or 1) multiplied by the place value of its position. **DAT-1.C.5** The place value of each position is determined by the base raised to the power of the position. Positions are numbered starting at the rightmost position with 0 and increasing by 1 for each subsequent position to the left.

continued on next page

© 2023 College Board

Big Idea 2: Data (DAT) *(cont'd)*

Enduring Understanding	Learning Objective	Essential Knowledge
DAT-1 *The way a computer represents data internally is different from the way the data are interpreted and displayed for the user. Programs are used to translate data into a representation more easily understood by people.*	**DAT-1.D** Compare data compression algorithms to determine which is best in a particular context. 1.D	**DAT-1.D.1** *Data compression* can reduce the size (number of bits) of transmitted or stored data. **DAT-1.D.2** Fewer bits does not necessarily mean less information. **DAT-1.D.3** The amount of size reduction from compression depends on both the amount of redundancy in the original data representation and the compression algorithm applied. **DAT-1.D.4** *Lossless* data compression algorithms can usually reduce the number of bits stored or transmitted while guaranteeing complete reconstruction of the original data. **DAT-1.D.5** *Lossy* data compression algorithms can significantly reduce the number of bits stored or transmitted but only allow reconstruction of an approximation of the original data. **DAT-1.D.6** Lossy data compression algorithms can usually reduce the number of bits stored or transmitted more than lossless compression algorithms. **DAT-1.D.7** In situations where quality or ability to reconstruct the original is maximally important, lossless compression algorithms are typically chosen. **DAT-1.D.8** In situations where minimizing data size or transmission time is maximally important, lossy compression algorithms are typically chosen.
DAT-2 *Programs can be used to process data, which allows users to discover information and create new knowledge.*	**DAT-2.A** Describe what information can be extracted from data. 5.B	**DAT-2.A.1** *Information* is the collection of facts and patterns extracted from data. **DAT-2.A.2** Data provide opportunities for identifying trends, making connections, and addressing problems. **DAT-2.A.3** Digitally processed data may show correlation between variables. A correlation found in data does not necessarily indicate that a causal relationship exists. Additional research is needed to understand the exact nature of the relationship. **DAT-2.A.4** Often, a single source does not contain the data needed to draw a conclusion. It may be necessary to combine data from a variety of sources to formulate a conclusion.

continued on next page

© 2023 College Board

Big Idea 2: Data (DAT) *(cont'd)*

Enduring Understanding	Learning Objective	Essential Knowledge
DAT-2 *Programs can be used to process data, which allows users to discover information and create new knowledge.*	**DAT-2.B** Describe what information can be extracted from metadata. 5.B	**DAT-2.B.1** *Metadata* are data about data. For example, the piece of *data* may be an image, while the *metadata* may include the date of creation or the file size of the image.
		DAT-2.B.2 Changes and deletions made to metadata do not change the primary data.
		DAT-2.B.3 Metadata are used for finding, organizing, and managing information.
		DAT-2.B.4 Metadata can increase the effective use of data or data sets by providing additional information.
		DAT-2.B.5 Metadata allow data to be structured and organized.
	DAT-2.C Identify the challenges associated with processing data. 5.D	**DAT-2.C.1** The ability to process data depends on the capabilities of the users and their tools.
		DAT-2.C.2 Data sets pose challenges regardless of size, such as: ▪ the need to clean data ▪ incomplete data ▪ invalid data ▪ the need to combine data sources
		DAT-2.C.3 Depending on how data were collected, they may not be uniform. For example, if users enter data into an open field, the way they choose to abbreviate, spell, or capitalize something may vary from user to user.
		DAT-2.C.4 *Cleaning data* is a process that makes the data uniform without changing their meaning (e.g., replacing all equivalent abbreviations, spellings, and capitalizations with the same word).
		DAT-2.C.5 Problems of bias are often created by the type or source of data being collected. Bias is not eliminated by simply collecting more data.
		DAT-2.C.6 The size of a data set affects the amount of information that can be extracted from it.
		DAT-2.C.7 Large data sets are difficult to process using a single computer and may require parallel systems.
		DAT-2.C.8 Scalability of systems is an important consideration when working with data sets, as the computational capacity of a system affects how data sets can be processed and stored.

continued on next page

© 2023 College Board

Big Idea 2: Data (DAT) *(cont'd)*

Enduring Understanding	Learning Objective	Essential Knowledge
DAT-2 *Programs can be used to process data, which allows users to discover information and create new knowledge.*	**DAT-2.D** Extract information from data using a program. 2.B	**DAT-2.D.1** Programs can be used to process data to acquire information.
		DAT-2.D.2 Tables, diagrams, text, and other visual tools can be used to communicate insight and knowledge gained from data.
		DAT-2.D.3 Search tools are useful for efficiently finding information.
		DAT-2.D.4 Data filtering systems are important tools for finding information and recognizing patterns in data.
		DAT-2.D.5 Programs such as spreadsheets help efficiently organize and find trends in information.
		DAT-2.D.6 Some processes that can be used to extract or modify information from data include the following: ▪ transforming every element of a data set, such as doubling every element in a list, or adding a parent's email to every student record ▪ filtering a data set, such as keeping only the positive numbers from a list, or keeping only students who signed up for band from a record of all the students ▪ combining or comparing data in some way, such as adding up a list of numbers, or finding the student who has the highest GPA ▪ visualizing a data set through a chart, graph, or other visual representation
	DAT-2.E Explain how programs can be used to gain insight and knowledge from data. 5.B	**DAT-2.E.1** Programs are used in an iterative and interactive way when processing information to allow users to gain insight and knowledge about data.
		DAT-2.E.2 Programmers can use programs to filter and clean digital data, thereby gaining insight and knowledge.
		DAT-2.E.3 Combining data sources, clustering data, and classifying data are parts of the process of using programs to gain insight and knowledge from data.
		DAT-2.E.4 Insight and knowledge can be obtained from translating and transforming digitally represented information.
		DAT-2.E.5 Patterns can emerge when data are transformed using programs.

© 2023 College Board

Big Idea 3: Algorithms and Programming (AAP)

Programmers integrate algorithms and abstraction to create programs for creative purposes and to solve problems. Using multiple program statements in a specified order, making decisions, and repeating the same process multiple times are the building blocks of programs. Incorporating elements of abstraction, by breaking problems down into interacting pieces, each with their own purpose, makes writing complex programs easier. Programmers need to think algorithmically and use abstraction to define and interpret processes that are used in a program.

Enduring Understanding	Learning Objective	Essential Knowledge
AAP-1 *To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways.*	**AAP-1.A** Represent a value with a variable. 3.A	**AAP-1.A.1** A *variable* is an abstraction inside a program that can hold a value. Each variable has associated data storage that represents one value at a time, but that value can be a list or other collection that in turn contains multiple values. **AAP-1.A.2** Using meaningful variable names helps with the readability of program code and understanding of what values are represented by the variables. **AAP-1.A.3** Some programming languages provide *types* to represent data, which are referenced using variables. These types include numbers, Booleans, lists, and strings. **AAP-1.A.4** Some values are better suited to representation using one type of datum rather than another.

	AAP-1.B
	Determine the value of a variable as a result of an assignment. 4.B
		AAP-1.B.1
		The assignment operator allows a program to change the value represented by a variable.
		
		AAP-1.B.2
		The exam reference sheet provides the "←" operator to use for assignment. For example,
		Text:
		`a ← expression`
		Block:
		`a ← expression`
		evaluates `expression` and then assigns a copy of the result to the variable `a`.
		
		AAP-1.B.3
		The value stored in a variable will be the most recent value assigned. For example:
		`a ← 1`
		`b ← a`
		`a ← 2`
		`display(b)`
		still displays 1.

continued on next page

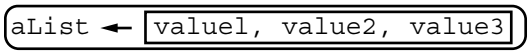
© 2023 College Board

Big Idea 3: Algorithms and Programming (AAP) *(cont'd)*

Enduring Understanding	Learning Objective	Essential Knowledge
AAP-1 *To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways.* *continued on next page*	**AAP-1.C** Represent a list or string using a variable. 3.A	**AAP-1.C.1** A *list* is an ordered sequence of elements. For example, `[value1, value2, value3, ...]` describes a list where `value1` is the first element, `value2` is the second element, `value3` is the third element, and so on. **AAP-1.C.2** An *element* is an individual value in a list that is assigned a unique index. **AAP-1.C.3** An *index* is a common method for referencing the elements in a list or string using natural numbers. **AAP-1.C.4** A *string* is an ordered sequence of characters.

© 2023 College Board

Enduring Understanding	Learning Objective	Essential Knowledge

AAP-1
To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways.
	AAP-1.D
	For data abstraction:
	a. Develop data abstraction using lists to store multiple elements. 3.B
	b. Explain how the use of data abstraction manages complexity in program code. 3.C
		AAP-1.D.1
		Data abstraction provides a separation between the abstract properties of a data type and the concrete details of its representation.
		
		AAP-1.D.2
		Data abstractions manage complexity in programs by giving a collection of data a name without referencing the specific details of the representation.
		
		AAP-1.D.3
		Data abstractions can be created using lists.
		
		AAP-1.D.4
		Developing a data abstraction to implement in a program can result in a program that is easier to develop and maintain.
		
		AAP-1.D.5
		Data abstractions often contain different types of elements.
		
		AAP-1.D.6
		The use of lists allows multiple related items to be treated as a single value. Lists are referred to by different names, such as *array,* depending on the programming language.
		
		X EXCLUSION STATEMENT (EK APP-1.D.6)
		The use of linked lists is outside the scope of this course and the AP Exam.
		
		AAP-1.D.7
		The exam reference sheet provides the notation
		`[value1, value2, value3, ...]`
		to create a list with those values as the first, second, third, and so on items. For example,
		▪ Text:
		`aList ← [value1, value2, value3, ...]`
		Block:
		 
		
		creates a new list that contains the values `value1`, `value2`, `value3`, and `...` at indices 1, 2, 3, and `...` respectively and assigns it to `aList`.
		▪ Text:
		`aList ← []`
		Block:
		
		creates a new empty list and assigns it to `aList`.

continued on next page

© 2023 College Board

Big Idea 3: Algorithms and Programming (AAP) *(cont'd)*

Enduring Understanding	Learning Objective	Essential Knowledge
AAP-1 *To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways.*		▪ Text: `aList ← bList` Block: assigns a copy of the list `bList` to the list `aList`. For example, if `bList` contains `[20, 40, 60]`, then `aList` will also contain `[20, 40, 60]` after the assignment.
		AAP-1.D.8 The exam reference sheet describes a list structure whose index values are 1 through the number of elements in the list, inclusive. For all list operations, if a list index is less than 1 or greater than the length of the list, an error message is produced and the program will terminate.
AAP-2 *The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.*	**AAP-2.A** Express an algorithm that uses sequencing without using a programming language. 2.A	**AAP-2.A.1** An *algorithm* is a finite set of instructions that accomplish a specific task.
		AAP-2.A.2 Beyond visual and textual programming languages, algorithms can be expressed in a variety of ways, such as natural language, diagrams, and pseudocode.
		AAP-2.A.3 Algorithms executed by programs are implemented using programming languages.
		AAP-2.A.4 Every algorithm can be constructed using combinations of sequencing, selection, and iteration.
	AAP-2.B Represent a step-by-step algorithmic process using sequential code statements. 2.B	**AAP-2.B.1** *Sequencing* is the application of each step of an algorithm in the order in which the code statements are given.
		AAP-2.B.2 A *code statement* is a part of program code that expresses an action to be carried out.
		AAP-2.B.3 An *expression* can consist of a value, a variable, an operator, or a procedure call that returns a value.
		AAP-2.B.4 Expressions are evaluated to produce a single value.
		AAP-2.B.5 The evaluation of expressions follows a set order of operations defined by the programming language.
		AAP-2.B.6 Sequential statements execute in the order they appear in the code segment.
		AAP-2.B.7 Clarity and readability are important considerations when expressing an algorithm in a programming language.

continued on next page

© 2023 College Board

Enduring Understanding	Learning Objective	Essential Knowledge
AAP-2 *The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.*	**AAP-2.C** Evaluate expressions that use arithmetic operators. 4.B	**AAP-2.C.1** Arithmetic operators are part of most programming languages and include addition, subtraction, multiplication, division, and modulus operators.
		AAP-2.C.2 The exam reference sheet provides `a MOD b`, which evaluates to the remainder when `a` is divided by `b`. Assume that `a` is an integer greater than or equal to 0 and `b` is an integer greater than `0`. For example, `17 MOD 5` evaluates to `2`.
		AAP-2.C.3 The exam reference sheet provides the arithmetic operators `+`, `–`, `*`, `/`, and `MOD`. Text and Block: ▪ `a + b` ▪ `a – b` ▪ `a * b` ▪ `a / b` ▪ `a MOD b` These are used to perform arithmetic on `a` and `b`. For example, `17 / 5` evaluates to `3.4`.
		AAP-2.C.4 The order of operations used in mathematics applies when evaluating expressions. The `MOD` operator has the same precedence as the `*` and `/` operators.
	AAP-2.D Evaluate expressions that manipulate strings. 4.B	**AAP-2.D.1** *String concatenation* joins together two or more strings end-to-end to make a new string.
		AAP-2.D.2 A *substring* is part of an existing string.
	AAP-2.E For relationships between two variables, expressions, or values: a. Write expressions using relational operators. 2.B b. Evaluate expressions that use relational operators. 4.B	**AAP-2.E.1** A *Boolean value* is either true or false.
		AAP-2.E.2 The exam reference sheet provides the following relational operators: `=`, `≠`, `>`, `<`, `≥`, and `≤`. Text and Block: ▪ `a = b` ▪ `a ≠ b` ▪ `a > b` ▪ `a < b` ▪ `a ≥ b` ▪ `a ≤ b` These are used to test the relationship between two variables, expressions, or values. A comparison using a relational operator evaluates to a Boolean value. For example, `a = b` evaluates to `true` if `a` and `b` are equal; otherwise, it evaluates to `false`.

continued on next page

© 2023 College Board

Enduring Understanding	Learning Objective	Essential Knowledge
AAP-2 *The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.*	**AAP-2.F** For relationships between Boolean values: a. Write expressions using logical operators. 2.B b. Evaluate expressions that use logic operators. 4.B	**AAP-2.F.1** The exam reference sheet provides the logical operators `NOT`, `AND`, and `OR`, which evaluate to a Boolean value.

		AAP-2.F.2
		The exam reference sheet provides
		Text:
		`NOT condition`
		Block:
		`NOT (condition)`
		which evaluates to `true` if `condition` is `false`; otherwise it evaluates to `false`.

		AAP-2.F.3
		The exam reference sheet provides
		Text:
		`condition1 AND condition2`
		Block:
		`(condition1) AND (condition2)`
		which evaluates to `true` if both `condition1` and `condition2` are `true`; otherwise it evaluates to `false`.

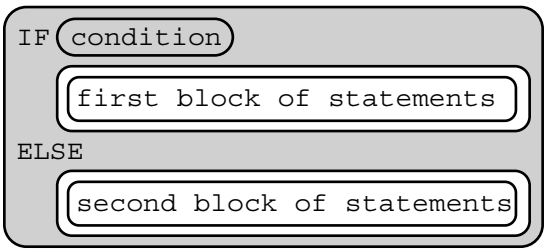
		AAP-2.F.4
		The exam reference sheet provides
		Text:
		`condition1 OR condition2`
		Block:
		`(condition1) OR (condition2)`
		which evaluates to `true` if `condition1` is `true` or if `condition2` is `true` or if both `condition1` and `condition2` are `true`; otherwise it evaluates to `false`.

		AAP-2.F.5 The operand for a logical operator is either a Boolean expression or a single Boolean value.
	AAP-2.G Express an algorithm that uses selection without using a programming language. 2.A	**AAP-2.G.1** *Selection* determines which parts of an algorithm are executed based on a condition being `true` or `false`.

continued on next page

© 2023 College Board

Enduring Understanding	Learning Objective	Essential Knowledge

AAP-2
The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.
	AAP-2.H
	For selection:
	a. Write conditional statements. 2.B
	b. Determine the result of conditional statements. 4.B
		AAP-2.H.1
		Conditional statements, or "if-statements," affect the sequential flow of control by executing different statements based on the value of a Boolean expression.
		
		AAP-2.H.2
		The exam reference sheet provides
		Text:
		`IF(condition)`
		`{`
		`  <block of statements>`
		`}`
		Block:
		
		in which the code in `block of statements` is executed if the Boolean expression `condition` evaluates to `true`; no action is taken if `condition` evaluates to `false`.
		
		AAP-2.H.3
		The exam reference sheet provides
		Text:
		`IF(condition)`
		`{`
		`  <first block of statements>`
		`}`
		`ELSE`
		`{`
		`  <second block of statements>`
		`}`
		Block:
		 
		
		in which the code in `first block of statements` is executed if the Boolean expression `condition` evaluates to `true`; otherwise, the code in `second block of statements` is executed.

	AAP-2.I For nested selection: a. Write nested conditional statements. 2.B b. Determine the result of nested conditional statements. 4.B	**AAP-2.I.1** Nested conditional statements consist of conditional statements within conditional statements.

continued on next page

© 2023 College Board

Enduring Understanding	Learning Objective	Essential Knowledge
AAP-2 *The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.*	**AAP-2.J** Express an algorithm that uses iteration without using a programming language. 2.A	**AAP-2.J.1** *Iteration* is a repeating portion of an algorithm. Iteration repeats a specified number of times or until a given condition is met.
	AAP-2.K For iteration: a. Write iteration statements. 2.B b. Determine the result or side effect of iteration statements. 4.B	**AAP-2.K.1** Iteration statements change the sequential flow of control by repeating a set of statements zero or more times, until a stopping condition is met.

		AAP-2.K.2
		The exam reference sheet provides
		Text:
		`REPEAT n TIMES`
		`{`
		`  <block of statements>`
		`}`
		Block:
		
		in which the `block of statements` is executed `n` times.

		AAP-2.K.3
		The exam reference sheet provides
		Text:
		`REPEAT UNTIL(condition)`
		`{`
		`  <block of statements>`
		`}`
		Block:
		
		in which the code in `block of statements` is repeated until the Boolean expression `condition` evaluates to `true`.

		AAP-2.K.4 In `REPEAT UNTIL(condition)` iteration, an infinite loop occurs when the ending condition will never evaluate to `true`.
		AAP-2.K.5 In `REPEAT UNTIL(condition)` iteration, if the conditional evaluates to `true` initially, the loop body is not executed at all, due to the condition being checked before the loop.

continued on next page

© 2023 College Board

Big Idea 3: Algorithms and Programming (AAP) *(cont'd)*

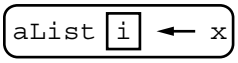
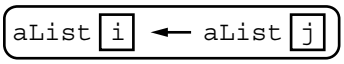
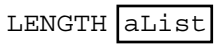
Enduring Understanding	Learning Objective	Essential Knowledge
AAP-2 *The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.*	**AAP-2.L** Compare multiple algorithms to determine if they yield the same side effect or result. 1.D	**AAP-2.L.1** Algorithms can be written in different ways and still accomplish the same tasks. **AAP-2.L.2** Algorithms that appear similar can yield different side effects or results. **AAP-2.L.3** Some conditional statements can be written as equivalent Boolean expressions. **AAP-2.L.4** Some Boolean expressions can be written as equivalent conditional statements. **AAP-2.L.5** Different algorithms can be developed or used to solve the same problem.
	AAP-2.M For algorithms: a. Create algorithms. 2.A b. Combine and modify existing algorithms. 2.B	**AAP-2.M.1** Algorithms can be created from an idea, by combining existing algorithms, or by modifying existing algorithms. **AAP-2.M.2** Knowledge of existing algorithms can help in constructing new ones. Some existing algorithms include: ▪ determining the maximum or minimum value of two or more numbers ▪ computing the sum or average of two or more numbers ▪ identifying if an integer is or is not evenly divisible by another integer ▪ determining a robot's path through a maze **AAP-2.M.3** Using existing correct algorithms as building blocks for constructing another algorithm has benefits such as reducing development time, reducing testing, and simplifying the identification of errors.

	AAP-2.N
	For list operations:
	a. Write expressions that use list indexing and list procedures. 2.B
	b. Evaluate expressions that use list indexing and list procedures. 4.B
		AAP-2.N.1
		The exam reference sheet provides basic operations on lists, including:
		▪ accessing an element by index
		Text:
		`aList[i]`
		Block:
		`aList` [`i`]
		accesses the element of `aList` at index `i`. The first element of `aList` is at index `1` and is accessed using the notation `aList[1]`.
		▪ assigning a value of an element of a list to a variable
		Text:
		`x ← aList [i]`
		Block:
		`x` ← `aList` [`i`]
		assigns the value of `aList[i]` to the variable `x`.

continued on next page

© 2023 College Board

Enduring Understanding	Learning Objective	Essential Knowledge

AAP-2
The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.
		▪ assigning a value to an element of a list
		Text:
		`aList[i] ← x`
		Block:
		 
		
		assigns the value of `x` to `aList[i]`.
		Text:
		`aList[i] ← aList[j]`
		Block:
		 
		
		assigns the value of `aList[j]` to `aList[i]`.
		▪ inserting elements at a given index
		Text:
		`INSERT(aList, i, value)`
		Block:
		
		shifts to the right any values in `aList` at indices greater than or equal to `i`. The length of the list is increased by 1, and `value` is placed at index `i` in `aList`.
		▪ adding elements to the end of the list
		Text:
		`APPEND(aList, value)`
		Block:
		
		increases the length of `aList` by 1, and `value` is placed at the end of `aList`.
		▪ removing elements
		Text:
		`REMOVE(aList, i)`
		Block:
		
		removes the item at index `i` in `aList` and shifts to the left any values at indices greater than `i`. The length of `aList` is decreased by 1.
		▪ determining the length of a list
		Text:
		`LENGTH(aList)`
		Block:
		 
		
		evaluates to the number of elements currently in `aList`.

		AAP-2.N.2 List procedures are implemented in accordance with the syntax rules of the programming language.

continued on next page

© 2023 College Board

Big Idea 3: Algorithms and Programming (AAP) *(cont'd)*

Enduring Understanding	Learning Objective	Essential Knowledge

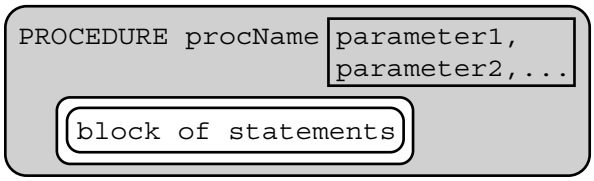
AAP-2
The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.
	AAP-2.O
	For algorithms involving elements of a list:
	a. Write iteration statements to traverse a list. 2.B
	b. Determine the result of an algorithm that includes list traversals. 4.B
		AAP-2.O.1
		Traversing a list can be a complete traversal, where all elements in the list are accessed, or a partial traversal, where only a portion of elements are accessed.
		EXCLUSION STATEMENT (EK AAP-2.O.1): *Traversing multiple lists at the same time using the same index for both (parallel traversals) is outside the scope of this course and the AP Exam.*
		AAP-2.O.2
		Iteration statements can be used to traverse a list.
		AAP-2.O.3
		The exam reference sheet provides
		Text:
		`FOR EACH item IN aList`
		`{`
		`  <block of statements>`
		`}`
		Block:
		`FOR EACH item IN aList`
		`block of statements`
		The variable `item` is assigned the value of each element of `aList` sequentially, in order, from the first element to the last element. The code in `block of statements` is executed once for each assignment of `item`.
		AAP-2.O.4
		Knowledge of existing algorithms that use iteration can help in constructing new algorithms. Some examples of existing algorithms that are often used with lists include:
		▪ determining a minimum or maximum value in a list
		▪ computing a sum or average of a list of numbers
		AAP-2.O.5
		Linear search or sequential search algorithms check each element of a list, in order, until the desired value is found or all elements in the list have been checked.

	AAP-2.P For binary search algorithms: a. Determine the number of iterations required to find a value in a data set. 1.D b. Explain the requirements necessary to complete a binary search. 1.A	**AAP-2.P.1** The binary search algorithm starts at the middle of a sorted data set of numbers and eliminates half of the data; this process repeats until the desired value is found or all elements have been eliminated. **EXCLUSION STATEMENT (EK: AAP-2.P.1):** *Specific implementations of the binary search are outside the scope of the course and the AP Exam.* **AAP-2.P.2** Data must be in sorted order to use the binary search algorithm. **AAP-2.P.3** Binary search is often more efficient than sequential/linear search when applied to sorted data.

continued on next page

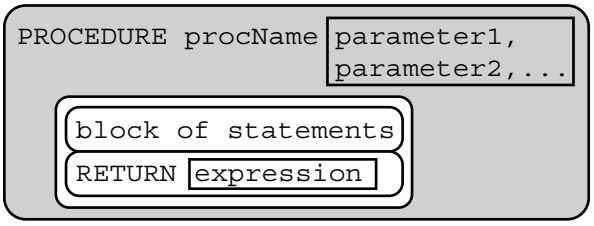
© 2023 College Board

Enduring Understanding	Learning Objective	Essential Knowledge

AAP-3
Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing them to write programs more quickly and with more confidence.
continued on next page
	AAP-3.A
	For procedure calls:
	a. Write statements to call procedures. 3.B
	b. Determine the result or effect of a procedure call. 4.B
		AAP-3.A.1
		A *procedure* is a named group of programming instructions that may have parameters and return values.
		
		AAP-3.A.2
		Procedures are referred to by different names, such as *method* or *function*, depending on the programming language.
		
		AAP-3.A.3
		Parameters are input variables of a procedure. *Arguments* specify the values of the parameters when a procedure is called.
		
		AAP-3.A.4
		A procedure call interrupts the sequential execution of statements, causing the program to execute the statements within the procedure before continuing. Once the last statement in the procedure (or a return statement) has executed, flow of control is returned to the point immediately following where the procedure was called.
		
		AAP-3.A.5
		The exam reference sheet provides
		`procName(arg1, arg2, ...)`
		as a way to call
		Text:
		`PROCEDURE procName(parameter1, parameter2, ...)`
		`{`
		`  <block of statements>`
		`}`
		Block:
		 
		which takes zero or more arguments; `arg1` is assigned to `parameter1`, `arg2` is assigned to `parameter2`, and so on.
		
		AAP-3.A.6
		The exam reference sheet provides the procedure
		Text:
		`DISPLAY(expression)`
		Block:
		
		to display the value of `expression`, followed by a space.

© 2023 College Board

Enduring Understanding	Learning Objective	Essential Knowledge

AAP-3
Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing them to write programs more quickly and with more confidence.
continued on next page
		AAP-3.A.7
		The exam reference sheet provides the
		Text:
		`RETURN(expression)`
		Block:
		
		statement, which is used to return the flow of control to the point where the procedure was called and to return the value of `expression`.
		
		AAP-3.A.8
		The exam reference sheet provides
		`result ← procName(arg1, arg2, ...)`
		to assign to `result` the "value of the procedure" being returned by calling
		Text:
		`PROCEDURE procName(parameter1, parameter2, ...)`
		`{`
		`<block of statements>`
		`RETURN(expression)`
		`}`
		Block:
		 
		
		AAP-3.A.9
		The exam reference sheet provides procedure
		Text:
		`INPUT()`
		Block:
		`INPUT`
		which accepts a value from the user and returns the input value.

© 2023 College Board

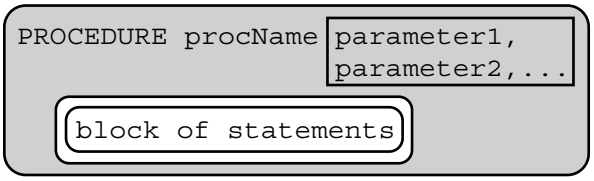
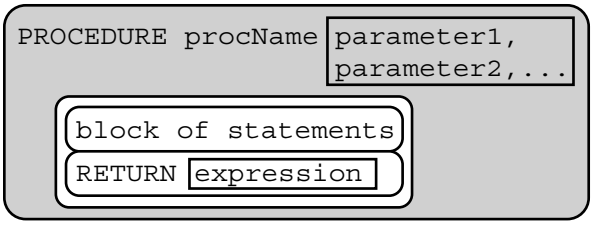
Big Idea 3: Algorithms and Programming (AAP) *(cont'd)*

Enduring Understanding	Learning Objective	Essential Knowledge
AAP-3 *Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing them to write programs more quickly and with more confidence.*	**AAP-3.B** Explain how the use of procedural abstraction manages complexity in a program. 3.C	**AAP-3.B.1** One common type of abstraction is procedural abstraction, which provides a name for a process and allows a procedure to be used only knowing what it does, not how it does it. **AAP-3.B.2** Procedural abstraction allows a solution to a large problem to be based on the solutions of smaller subproblems. This is accomplished by creating procedures to solve each of the subproblems. **AAP-3.B.3** The subdivision of a computer program into separate subprograms is called *modularity*. **AAP-3.B.4** A procedural abstraction may extract shared features to generalize functionality instead of duplicating code. This allows for program code reuse, which helps manage complexity. **AAP-3.B.5** Using parameters allows procedures to be generalized, enabling the procedures to be reused with a range of input values or arguments. **AAP-3.B.6** Using procedural abstraction helps improve code readability. **AAP-3.B.7** Using procedural abstraction in a program allows programmers to change the internals of the procedure (to make it faster, more efficient, use less storage, etc.) without needing to notify users of the change as long as what the procedure does is preserved.

continued on next page

© 2023 College Board

Enduring Understanding	Learning Objective	Essential Knowledge

AAP-3
Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing them to write programs more quickly and with more confidence.
	AAP-3.C
	Develop procedural abstractions to manage complexity in a program by writing procedures. 3.B
		AAP-3.C.1
		The exam reference sheet provides
		Text:
		`PROCEDURE procName(parameter1, parameter2, ...)`
		`{`
		`  <block of statements>`
		`}`
		Block:
		 
		
		which is used to define a procedure that takes zero or more arguments. The procedure contains `block of statements`.
		
		AAP-3.C.2
		The exam reference sheet provides
		Text:
		`PROCEDURE procName(parameter1, parameter2, ...)`
		`{`
		`  <block of statements>`
		`  RETURN(expression)`
		`}`
		Block:
		 
		
		which is used to define a procedure that takes zero or more arguments. The procedure contains `block of statements` and returns the value of `expression`. The `RETURN` statement may appear at any point inside the procedure and causes an immediate return from the procedure back to the calling statement.

continued on next page

© 2023 College Board

Big Idea 3: Algorithms and Programming (AAP) *(cont'd)*

Enduring Understanding	Learning Objective	Essential Knowledge
AAP-3 *Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing them to write programs more quickly and with more confidence.*	**AAP-3.D** Select appropriate libraries or existing code segments to use in creating new programs. 2.B	**AAP-3.D.1** A software library contains procedures that may be used in creating new programs. **AAP-3.D.2** Existing code segments can come from internal or external sources, such as libraries or previously written code. **AAP-3.D.3** The use of libraries simplifies the task of creating complex programs. **AAP-3.D.4** Application program interfaces (APIs) are specifications for how the procedures in a library behave and can be used. **AAP-3.D.5** Documentation for an API/library is necessary in understanding the behaviors provided by the API/library and how to use them.

	AAP-3.E
	For generating random values:
	a. Write expressions to generate possible values. 2.B
	b. Evaluate expressions to determine the possible results. 4.B
		AAP-3.E.1
		The exam reference sheet provides
		Text:
		`RANDOM(a, b)`
		Block:
		`RANDOM` `a, b`
		which generates and returns a random integer from `a` to `b`, inclusive. Each result is equally likely to occur. For example, `RANDOM(1, 3)` could return `1`, `2`, or `3`.
		AAP-3.E.2
		Using random number generation in a program means each execution may produce a different result.

continued on next page

© 2023 College Board

Enduring Understanding	Learning Objective	Essential Knowledge
AAP-3 *Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing them to write programs more quickly and with more confidence.*	**AAP-3.F** For simulations: a. Explain how computers can be used to represent real-world phenomena or outcomes. 1.A b. Compare simulations with real-world contexts. 1.D	**AAP-3.F.1** Simulations are abstractions of more complex objects or phenomena for a specific purpose. **AAP-3.F.2** A *simulation* is a representation that uses varying sets of values to reflect the changing state of a phenomenon. **AAP-3.F.3** Simulations often mimic real-world events with the purpose of drawing inferences, allowing investigation of a phenomenon without the constraints of the real world. **AAP-3.F.4** The process of developing an abstract simulation involves removing specific details or simplifying functionality. **AAP-3.F.5** Simulations can contain bias derived from the choices of real-world elements that were included or excluded. **AAP-3.F.6** Simulations are most useful when real-world events are impractical for experiments (e.g., too big, too small, too fast, too slow, too expensive, or too dangerous). **AAP-3.F.7** Simulations facilitate the formulation and refinement of hypotheses related to the objects or phenomena under consideration. **AAP-3.F.8** Random number generators can be used to simulate the variability that exists in the real world.
AAP-4 *There exist problems that computers cannot solve, and even when a computer can solve a problem, it may not be able to do so in a reasonable amount of time.*	**AAP-4.A** For determining the efficiency of an algorithm: a. Explain the difference between algorithms that run in reasonable time and those that do not. 1.D b. Identify situations where a heuristic solution may be more appropriate. 1.D	**AAP-4.A.1** A *problem* is a general description of a task that can (or cannot) be solved algorithmically. An *instance* of a problem also includes specific input. For example, sorting is a problem; sorting the list (2,3,1,7) is an instance of the problem. **AAP-4.A.2** A *decision problem* is a problem with a yes/no answer (e.g., is there a path from A to B?). An *optimization problem* is a problem with the goal of finding the "best" solution among many (e.g., what is the shortest path from A to B?). **AAP-4.A.3** *Efficiency* is an estimation of the amount of computational resources used by an algorithm. Efficiency is typically expressed as a function of the size of the input. **X EXCLUSION STATEMENT (EK AAP-4.A.3):** *Formal analysis of algorithms (Big-O) and formal reasoning using mathematical formulas are outside the scope of this course and the AP Exam.* **AAP-4.A.4** An algorithm's efficiency is determined through formal or mathematical reasoning.

continued on next page

© 2023 College Board

Big Idea 3: Algorithms and Programming (AAP) *(cont'd)*

Enduring Understanding	Learning Objective	Essential Knowledge
AAP-4 *There exist problems that computers cannot solve, and even when a computer can solve a problem, it may not be able to do so in a reasonable amount of time.*		**AAP-4.A.5** An algorithm's efficiency can be informally measured by determining the number of times a statement or group of statements executes. **AAP-4.A.6** Different correct algorithms for the same problem can have different efficiencies. **AAP-4.A.7** Algorithms with a polynomial efficiency or lower (constant, linear, square, cube, etc.) are said to run in a *reasonable amount of time*. Algorithms with exponential or factorial efficiencies are examples of algorithms that run in an *unreasonable amount of time*. **AAP-4.A.8** Some problems cannot be solved in a reasonable amount of time because there is no efficient algorithm for solving them. In these cases, approximate solutions are sought. **AAP-4.A.9** A *heuristic* is an approach to a problem that produces a solution that is not guaranteed to be optimal but may be used when techniques that are guaranteed to always find an optimal solution are impractical. **EXCLUSION STATEMENT (AAP-4.A.9):** *Specific heuristic solutions are outside the scope of this course and the AP Exam.*
	AAP-4.B Explain the existence of undecidable problems in computer science. 1.A	**AAP-4.B.1** A *decidable problem* is a decision problem for which an algorithm can be written to produce a correct output for all inputs (e.g., "Is the number even?"). **AAP-4.B.2** An *undecidable problem* is one for which no algorithm can be constructed that is always capable of providing a correct yes-or-no answer. **EXCLUSION STATEMENT (EK AAP-4.B.2):** *Determining whether a given problem is undecidable is outside the scope of this course and the AP Exam.* **AAP-4.B.3** An undecidable problem may have some instances that have an algorithmic solution, but there is no algorithmic solution that could solve all instances of the problem.

© 2023 College Board

Big Idea 4: Computing Systems and Networks (CSN)

Computer systems and networks are used to transfer data. One of the largest and most commonly used networks is the Internet. Through a series of protocols, the Internet can be used to send and receive information and ideas throughout the world. Transferring and processing information can be slow when done on a single computer but leveraging multiple computers to do the work at the same time can significantly shorten the time it takes to complete tasks or solve problems.

Enduring Understanding	Learning Objective	Essential Knowledge
CSN-1 *Computer systems and networks facilitate the transfer of data.*	**CSN-1.A** Explain how computing devices work together in a network. 5.A	**CSN-1.A.1** A *computing device* is a physical artifact that can run a program. Some examples include computers, tablets, servers, routers, and smart sensors.
		CSN-1.A.2 A *computing system* is a group of computing devices and programs working together for a common purpose.
		CSN-1.A.3 A *computer network* is a group of interconnected computing devices capable of sending or receiving data.
		CSN-1.A.4 A computer network is a type of computing system.
		CSN-1.A.5 A *path* between two computing devices on a computer network (a sender and a receiver) is a sequence of directly connected computing devices that begins at the sender and ends at the receiver.
		CSN-1.A.6 *Routing* is the process of finding a path from sender to receiver.
		CSN-1.A.7 The *bandwidth* of a computer network is the maximum amount of data that can be sent in a fixed amount of time.
		CSN-1.A.8 Bandwidth is usually measured in bits per second.
	CSN-1.B Explain how the Internet works. 5.A	**CSN-1.B.1** The Internet is a computer network consisting of interconnected networks that use standardized, open (nonproprietary) communication protocols.
		CSN-1.B.2 Access to the Internet depends on the ability to connect a computing device to an Internet-connected device.
		CSN-1.B.3 A *protocol* is an agreed-upon set of rules that specify the behavior of a system.
		CSN-1.B.4 The protocols used in the Internet are *open*, which allows users to easily connect additional computing devices to the Internet.

continued on next page

© 2023 College Board

Big Idea 4: Computing Systems and Networks (CSN) *(cont'd)*

Enduring Understanding	Learning Objective	Essential Knowledge
CSN-1 *Computer systems and networks facilitate the transfer of data.*		**CSN-1.B.5** Routing on the Internet is usually dynamic; it is not specified in advance.
		CSN-1.B.6 The *scalability* of a system is the capacity for the system to change in size and scale to meet new demands.
		CSN-1.B.7 The Internet was designed to be scalable.
	CSN-1.C Explain how data are sent through the Internet via packets. 5.A	**CSN-1.C.1** Information is passed through the Internet as a *data stream*. Data streams contain chunks of data, which are encapsulated in *packets*.
		CSN-1.C.2 Packets contain a chunk of data and metadata used for routing the packet between the origin and the destination on the Internet, as well as for data reassembly.
		CSN-1.C.3 Packets may arrive at the destination in order, out of order, or not at all.
		CSN-1.C.4 IP, TCP, and UDP are common protocols used on the Internet.
	CSN-1.D Describe the differences between the Internet and the World Wide Web. 5.A	**CSN-1.D.1** The World Wide Web is a system of linked pages, programs, and files.
		CSN-1.D.2 HTTP is a protocol used by the World Wide Web.
		CSN-1.D.3 The World Wide Web uses the Internet.

continued on next page

© 2023 College Board

Big Idea 4: Computing Systems and Networks (CSN) *(cont'd)*

Enduring Understanding	Learning Objective	Essential Knowledge
CSN-1 *Computer systems and networks facilitate the transfer of data.* *continued on next page*	**CSN-1.E** For fault-tolerant systems, like the Internet: a. Describe the benefits of fault tolerance. 1.D b. Explain how a given system is fault-tolerant. 5.A c. Identify vulnerabilities to failure in a system. 1.D	**CSN-1.E.1** The Internet has been engineered to be fault-tolerant, with abstractions for routing and transmitting data. **CSN-1.E.2** *Redundancy* is the inclusion of extra components that can be used to mitigate failure of a system if other components fail. **CSN-1.E.3** One way to accomplish network redundancy is by having more than one path between any two connected devices. **CSN-1.E.4** If a particular device or connection on the Internet fails, subsequent data will be sent via a different route, if possible. **CSN-1.E.5** When a system can support failures and still continue to function, it is called *fault-tolerant*. This is important because elements of complex systems fail at unexpected times, often in groups, and fault tolerance allows users to continue to use the network. **CSN-1.E.6** Redundancy within a system often requires additional resources but can provide the benefit of fault tolerance. **CSN-1.E.7** The redundancy of routing options between two points increases the reliability of the Internet and helps it scale to more devices and more people.

© 2023 College Board

Big Idea 4: Computing Systems and Networks (CSN) *(cont'd)*

Enduring Understanding	Learning Objective	Essential Knowledge
CSN-2 *Parallel and distributed computing leverage multiple computers to more quickly solve complex problems or process large data sets.*	**CSN-2.A** For sequential, parallel, and distributed computing: a. Compare problem solutions. 1.D b. Determine the efficiency of solutions. 1.D	**CSN-2.A.1** *Sequential computing* is a computational model in which operations are performed in order one at a time. **CSN-2.A.2** *Parallel computing* is a computational model where the program is broken into multiple smaller sequential computing operations, some of which are performed simultaneously. **CSN-2.A.3** *Distributed computing* is a computational model in which multiple devices are used to run a program. **CSN-2.A.4** Comparing efficiency of solutions can be done by comparing the time it takes them to perform the same task. **CSN-2.A.5** A sequential solution takes as long as the sum of all of its steps. **CSN-2.A.6** A parallel computing solution takes as long as its sequential tasks plus the longest of its parallel tasks. **CSN-2.A.7** The "speedup" of a parallel solution is measured in the time it took to complete the task sequentially divided by the time it took to complete the task when done in parallel.
	CSN-2.B Describe benefits and challenges of parallel and distributed computing. 1.D	**CSN-2.B.1** Parallel computing consists of a parallel portion and a sequential portion. **CSN-2.B.2** Solutions that use parallel computing can scale more effectively than solutions that use sequential computing. **CSN-2.B.3** Distributed computing allows problems to be solved that could not be solved on a single computer because of either the processing time or storage needs involved. **CSN-2.B.4** Distributed computing allows much larger problems to be solved quicker than they could be solved using a single computer. **CSN-2.B.5** When increasing the use of parallel computing in a solution, the efficiency of the solution is still limited by the sequential portion. This means that at some point, adding parallel portions will no longer meaningfully increase efficiency.

© 2023 College Board

Big Idea 5: Impact of Computing (IOC)

Computers and computing have revolutionized our lives. To use computing safely and responsibly, we need to be aware of privacy, security, and ethical issues. As programmers, we need to understand the potential impacts of our programs and be responsible for the consequences. As computer users, we need to understand any potential beneficial or harmful effects and how to protect ourselves and our privacy when using a computer.

Enduring Understanding	Learning Objective	Essential Knowledge
IOC-1 *While computing innovations are typically designed to achieve a specific purpose, they may have unintended consequences.*	**IOC-1.A** Explain how an effect of a computing innovation can be both beneficial and harmful. 5.C	**IOC-1.A.1** People create computing innovations.
		IOC-1.A.2 The way people complete tasks often changes to incorporate new computing innovations.
		IOC-1.A.3 Not every effect of a computing innovation is anticipated in advance.
		IOC-1.A.4 A single effect can be viewed as both beneficial and harmful by different people, or even by the same person.
		IOC-1.A.5 Advances in computing have generated and increased creativity in other fields, such as medicine, engineering, communications, and the arts.
	IOC-1.B Explain how a computing innovation can have an impact beyond its intended purpose. 5.C	**IOC-1.B.1** Computing innovations can be used in ways that their creators had not originally intended: ▪ The World Wide Web was originally intended only for rapid and easy exchange of information within the scientific community. ▪ Targeted advertising is used to help businesses, but it can be misused at both individual and aggregate levels. ▪ Machine learning and data mining have enabled innovation in medicine, business, and science, but information discovered in this way has also been used to discriminate against groups of individuals.
		IOC-1.B.2 Some of the ways computing innovations can be used may have a harmful impact on society, the economy, or culture.
		IOC-1.B.3 Responsible programmers try to consider the unintended ways their computing innovations can be used and the potential beneficial and harmful effects of these new uses.
		IOC-1.B.4 It is not possible for a programmer to consider all the ways a computing innovation can be used.
		IOC-1.B.5 Computing innovations have often had unintended beneficial effects by leading to advances in other fields.
		IOC-1.B.6 Rapid sharing of a program or running a program with a large number of users can result in significant impacts beyond the intended purpose or control of the programmer.

continued on next page

© 2023 College Board

Big Idea 5: Impact of Computing (IOC) *(cont'd)*

Enduring Understanding	Learning Objective	Essential Knowledge
IOC-1 *While computing innovations are typically designed to achieve a specific purpose, they may have unintended consequences.*	**IOC-1.C** Describe issues that contribute to the digital divide. 5.C	**IOC-1.C.1** Internet access varies between socioeconomic, geographic, and demographic characteristics, as well as between countries.
		IOC-1.C.2 The "digital divide" refers to differing access to computing devices and the Internet, based on socioeconomic, geographic, or demographic characteristics.
		IOC-1.C.3 The digital divide can affect both groups and individuals.
		IOC-1.C.4 The digital divide raises issues of equity, access, and influence, both globally and locally.
		IOC-1.C.5 The digital divide is affected by the actions of individuals, organizations, and governments.
	IOC-1.D Explain how bias exists in computing innovations. 5.E	**IOC-1.D.1** Computing innovations can reflect existing human biases because of biases written into the algorithms or biases in the data used by the innovation.
		IOC-1.D.2 Programmers should take action to reduce bias in algorithms used for computing innovations as a way of combating existing human biases.
		IOC-1.D.3 Biases can be embedded at all levels of software development.
	IOC-1.E Explain how people participate in problem-solving processes at scale. 1.C	**IOC-1.E.1** Widespread access to information and public data facilitates the identification of problems, development of solutions, and dissemination of results.
		IOC-1.E.2 Science has been affected by using distributed and "citizen science" to solve scientific problems.
		IOC-1.E.3 Citizen science is scientific research conducted in whole or part by distributed individuals, many of whom may not be scientists, who contribute relevant data to research using their own computing devices.
		IOC-1.E.4 *Crowdsourcing* is the practice of obtaining input or information from a large number of people via the Internet.
		IOC-1.E.5 Human capabilities can be enhanced by collaboration via computing.
		IOC-1.E.6 Crowdsourcing offers new models for collaboration, such as connecting businesses or social causes with funding.

continued on next page

© 2023 College Board

Big Idea 5: Impact of Computing (IOC) *(cont'd)*

Enduring Understanding	Learning Objective	Essential Knowledge
IOC-1 *While computing innovations are typically designed to achieve a specific purpose, they may have unintended consequences.*	**IOC-1.F** Explain how the use of computing can raise legal and ethical concerns. 5.E	**IOC-1.F.1** Material created on a computer is the intellectual property of the creator or an organization. **IOC-1.F.2** Ease of access and distribution of digitized information raises intellectual property concerns regarding ownership, value, and use. **IOC-1.F.3** Measures should be taken to safeguard intellectual property. **IOC-1.F.4** The use of material created by someone else without permission and presented as one's own is plagiarism and may have legal consequences. **IOC-1.F.5** Some examples of legal ways to use materials created by someone else include: ▪ Creative Commons—a public copyright license that enables the free distribution of an otherwise copyrighted work. This is used when the content creator wants to give others the right to share, use, and build upon the work they have created. ▪ open source—programs that are made freely available and may be redistributed and modified ▪ open access—online research output free of any and all restrictions on access and free of many restrictions on use, such as copyright or license restrictions **IOC-1.F.6** The use of material created by someone other than you should always be cited. **IOC-1.F.7** Creative Commons, open source, and open access have enabled broad access to digital information. **IOC-1.F.8** As with any technology or medium, using computing to harm individuals or groups of people raises legal and ethical concerns. **IOC-1.F.9** Computing can play a role in social and political issues, which in turn often raises legal and ethical concerns. **IOC-1.F.10** The digital divide raises ethical concerns around computing. **IOC-1.F.11** Computing innovations can raise legal and ethical concerns. Some examples of these include: ▪ the development of software that allows access to digital media downloads and streaming ▪ the development of algorithms that include bias ▪ the existence of computing devices that collect and analyze data by continuously monitoring activities

continued on next page

© 2023 College Board

Big Idea 5: Impact of Computing (IOC) *(cont'd)*

Enduring Understanding	Learning Objective	Essential Knowledge
IOC-2 *The use of computing innovations may involve risks to personal safety and identity.*	**IOC-2.A** Describe the risks to privacy from collecting and storing personal data on a computer system. 5.D	**IOC-2.A.1** Personally identifiable information (PII) is information about an individual that identifies, links, relates, or describes them. Examples of PII include: ▪ Social Security number ▪ age ▪ race ▪ phone number(s) ▪ medical information ▪ financial information ▪ biometric data
		IOC-2.A.2 Search engines can record and maintain a history of searches made by users.
		IOC-2.A.3 Websites can record and maintain a history of individuals who have viewed their pages.
		IOC-2.A.4 Devices, websites, and networks can collect information about a user's location.
		IOC-2.A.5 Technology enables the collection, use, and exploitation of information about, by, and for individuals, groups, and institutions.
		IOC-2.A.6 Search engines can use search history to suggest websites or for targeted marketing.
		IOC-2.A.7 Disparate personal data, such as geolocation, cookies, and browsing history, can be aggregated to create knowledge about an individual.
		IOC-2.A.8 PII and other information placed online can be used to enhance a user's online experiences.
		IOC-2.A.9 PII stored online can be used to simplify making online purchases.
		IOC-2.A.10 Commercial and governmental curation of information may be exploited if privacy and other protections are ignored.
		IOC-2.A.11 Information placed online can be used in ways that were not intended and that may have a harmful impact. For example, an email message may be forwarded, tweets can be retweeted, and social media posts can be viewed by potential employers.
		IOC-2.A.12 PII can be used to stalk or steal the identity of a person or to aid in the planning of other criminal acts.

continued on next page

© 2023 College Board

Enduring Understanding	Learning Objective	Essential Knowledge
IOC-2 *The use of computing innovations may involve risks to personal safety and identity.*		**IOC-2.A.13** Once information is placed online, it is difficult to delete.
		IOC-2.A.14 Programs can collect your location and record where you have been, how you got there, and how long you were at a given location.
		IOC-2.A.15 Information posted to social media services can be used by others. Combining information posted on social media and other sources can be used to deduce private information about you.
	IOC-2.B Explain how computing resources can be protected and can be misused. 5.E	**IOC-2.B.1** Authentication measures protect devices and information from unauthorized access. Examples of authentication measures include strong passwords and multifactor authentication.
		IOC-2.B.2 A strong password is something that is easy for a user to remember but would be difficult for someone else to guess based on knowledge of that user.
		IOC-2.B.3 *Multifactor authentication* is a method of computer access control in which a user is only granted access after successfully presenting several separate pieces of evidence to an authentication mechanism, typically in at least two of the following categories: knowledge (something they know), possession (something they have), and inherence (something they are).
		IOC-2.B.4 Multifactor authentication requires at least two steps to unlock protected information; each step adds a new layer of security that must be broken to gain unauthorized access.
		IOC-2.B.5 *Encryption* is the process of encoding data to prevent unauthorized access. *Decryption* is the process of decoding the data. Two common encryption approaches are: ▪ Symmetric key encryption involves one key for both encryption and decryption. ▪ Public key encryption pairs a public key for encryption and a private key for decryption. The sender does not need the receiver's private key to encrypt a message, but the receiver's private key is required to decrypt the message. **X EXCLUSION STATEMENT (EK IOC-2.B.5):** *Specific mathematical procedures for encryption and decryption are beyond the scope of this course and the AP Exam.*
		IOC-2.B.6 Certificate authorities issue digital certificates that validate the ownership of encryption keys used in secure communications and are based on a trust model.

continued on next page

© 2023 College Board

Big Idea 5: Impact of Computing (IOC) *(cont'd)*

Enduring Understanding	Learning Objective	Essential Knowledge
IOC-2 *The use of computing innovations may involve risks to personal safety and identity.*		**IOC-2.B.7** Computer virus and malware scanning software can help protect a computing system against infection.
		IOC-2.B.8 A *computer virus* is a malicious program that can copy itself and gain access to a computer in an unauthorized way. Computer viruses often attach themselves to legitimate programs and start running independently on a computer.
		IOC-2.B.9 *Malware* is software intended to damage a computing system or to take partial control over its operation.
		IOC-2.B.10 All real-world systems have errors or design flaws that can be exploited to compromise them. Regular software updates help fix errors that could compromise a computing system.
		IOC-2.B.11 Users can control the permissions programs have for collecting user information. Users should review the permission settings of programs to protect their privacy.
	IOC-2.C Explain how unauthorized access to computing resources is gained. **5.E**	**IOC-2.C.1** *Phishing* is a technique that attempts to trick a user into providing personal information. That personal information can then be used to access sensitive online resources, such as bank accounts and emails.
		IOC-2.C.2 *Keylogging* is the use of a program to record every keystroke made by a computer user in order to gain fraudulent access to passwords and other confidential information.
		IOC-2.C.3 Data sent over public networks can be intercepted, analyzed, and modified. One way that this can happen is through a rogue access point.
		IOC-2.C.4 A *rogue access point* is a wireless access point that gives unauthorized access to secure networks.
		IOC-2.C.5 A malicious link can be disguised on a web page or in an email message.
		IOC-2.C.6 Unsolicited emails, attachments, links, and forms in emails can be used to compromise the security of a computing system. These can come from unknown senders or from known senders whose security has been compromised.
		IOC-2.C.7 Untrustworthy (often free) downloads from freeware or shareware sites can contain malware.

© 2023 College Board

© 2023 The College Board. 01553-093