AP Exam Preparation

AP Exam

When: Monday June 7, 2021 at 4PM Eastern Standard Time.

At home. On Digital Testing App.

Preparing:

1) Download! https://download.app.collegeboard.org/

2) Practice with Example Questions in the Digital Testing App.

3) 1–3 Days Before Exam Day, Complete Exam Setup for Each Digital Exam. You cannot take the exam without completing this step!

4) On Exam Day: 30 Minutes Before the Exam | Check In to the Exam

On exam day, you must check in **30 minutes before the official start time of the exam**—at 11:30 a.m. EDT for 12 p.m. exams and 3:30 p.m. EDT for 4 p.m. exams—to complete final pre-exam checks.

AP Exam

Section I: End-of-Course Multiple-Choice Exam

70 Multiple-Choice Questions | 120 Minutes | 70% of Score | 4 answer options

- 57 single-select multiple-choice
- 5 single-select with reading passage about a computing innovation
- 8 multiple-select multiple-choice: select 2 answers
- Note: On the digital exam, there are 59 single-select multiple-choice questions, 5 single-select multiple-choice questions with reading passage, and 6 multiple-select multiple-choice questions.

Section II: Create Performance Task

30% of Score

Topics

The AP Multiple Choice end-of-year covers 5 Big Ideas:

1) Creative Development
2) Data
3) Algorithms and Programming
4) Computer Systems and Networks
5) Impact of Computing

Big Idea 1

1.1 Collaboration	1.C Explain how collaboration affects the development of a solution. 6.A Collaborate in the development of solutions *(not assessed)*.
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.
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)*.
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.

AP Exam Review

Up to this point, I have been teaching Computer Science principles rather than teaching to the AP test.

This lecture slides is teaching to the AP test. It will provide review material for the AP test that we might not have covered during the course of the year.

We will go over each of the 5 ideas in summary.

Big Idea 1

Please read Chapter 2: Big Idea 1: Creative Development. of the AP Barron's book. The following slides provide a summary of the material covered in this chapter.

A **computing innovation** includes a program as an integral part of its function.

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).

Hardware is the physical components of a computing device, while software is the instructions in a programming language to the computing device. A computing innovation can have hardware components. However, the computing innovation is about the software, not the hardware.

Big Idea 1

Computing hardware has gotten smaller and more powerful over the years. Moore's law predicts that the size of transistors halves every two years while the cost also halves every two years. Computers went from taking up 1,800 square feet and weighing almost 50 tons to being able to fit in your pocket.

Software	Hardware
Operating systems	Motherboard
Driverless vehicle software to avoid crashes	Self-driving car
Dual-monitor programs for Windows	Monitor
Compiler	Transistor

Big Idea 1

Collaboration helps people learn from each other. Collaboration that includes diverse perspectives helps to avoid bias in the development of computing innovations.

For example, if females play video games at the same percentage as males, a game company might not avoid bias if it employed males to write the code for the games. Bringing in female coders could bring additional perspectives that might not have been achieved otherwise.

Programming companies often hire people who not only are good programmers but also have interpersonal skills needed to collaborate effectively. Effective collaboration can help one gain insight and knowledge by applying multiple perspectives, experiences, and skill sets.

Big Idea 1

Collaboration is a learned skill. That skill includes but is not limited to:

- Communication
- Consensus building
- Conflict resolution
- Negotiation

Collaboration with others can make the programmer more self-aware.

Group programming can match up your weaknesses with someone else's strengths, which results in a better product and leads to insight and knowledge not obtainable when working alone.

Big Idea 1

Collaboration is not limited by location. Current computing tools allow people in different physical locations to share data.

Online collaboration tools, such as Google Docs, Zoom, Slack, Yammer, and—by the time you read this—dozens of other tools, allow programmers to collaborate from home or from anywhere that has internet access.

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.

A **code segment** refers to a collection of program statements that are part of a program.

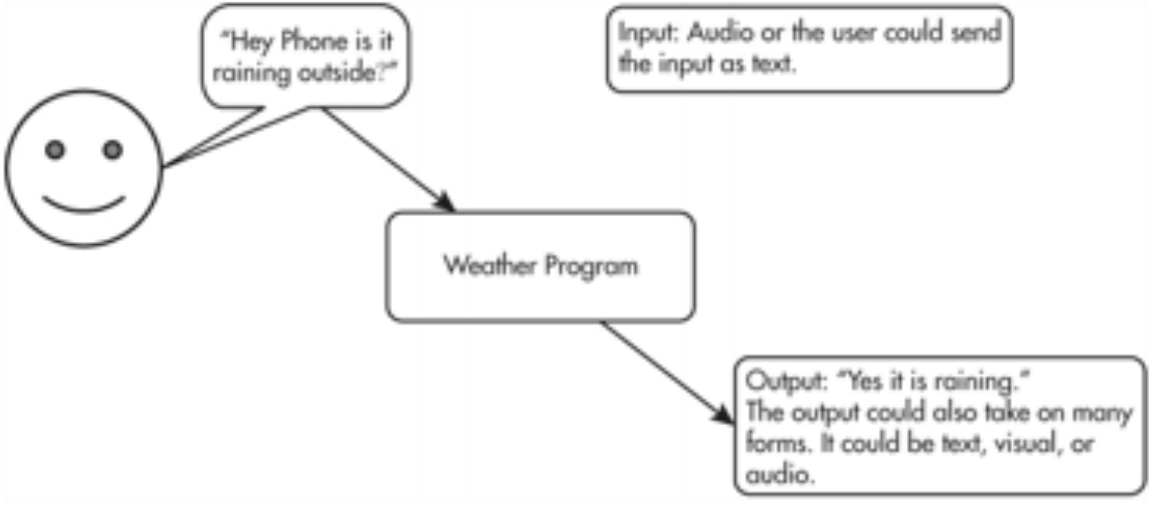
Big Idea 1

Program input is 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. For example, a cell phone can convert voice (audio) to text to send a message.

A weather program on your phone could take input in many forms. This weather app was triggered by the user saying (audio) "Hey Phone…," which would be an example of audio input.

This triggering is called an **event**. The event is the action that supplies input data to a program. Events can be generated when a key is pressed, a mouse is clicked, a program is started, or by any other defined action that affects the flow of execution.

Big Idea 1



Example of input/output

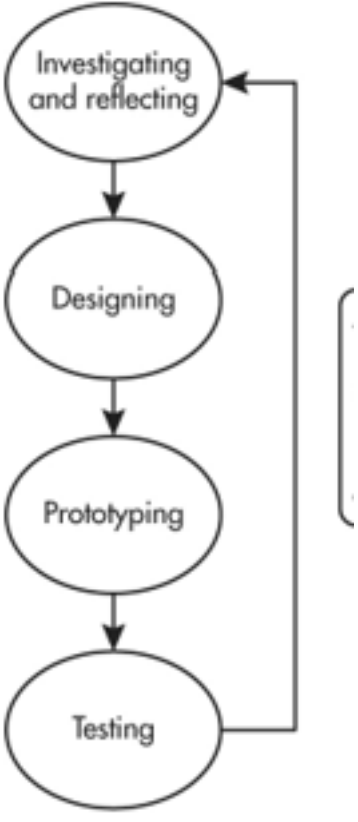
Program outputs are any data sent from a program to a device. Program output can come in a variety of forms, such as tactile, audio, visual, or text. Program output is usually based on a program's input or prior state (e.g., internal values).

Big Idea 1

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

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.

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.



Big Idea 1

The design of a program incorporates investigations to determine its requirements. Most programs are designed to be used by people other than the programmers.

To meet the needs of the users, the investigation must identify the program constraints as well as the concerns and interests of people who will use the program.

Some ways investigations can be performed are as follows:

- Collecting data through surveys
- User testing
- Interviews
- Direct observations

Big Idea 1

The design phase of a program may include:

- Brainstorming
- Planning and storyboarding
- Organizing the program into modules and functional components
- Creating diagrams that represent the layouts of the user interface
- Developing a testing strategy for the program

Program documentation is a written description of the function of a code segment, event, procedure, or program and how it was developed.

Program documentation helps in developing and maintaining correct programs when working individually or in collaborative programming environments.

Big Idea 1

Programmers should document a program throughout its development.

Documentation helps the programmer remember what he or she was thinking or the collaborative partners were thinking at the time they were programming.

Comments are a form of program documentation written into the program that do not affect how the program runs. Comments do not affect the run speed of a program. Python, for example, uses # for comments.

Big Idea 1

Three types of program errors can occur:

- **Logic error**—This is a mistake in the algorithm or program that causes it to behave incorrectly or unexpectedly. (incorrect implementation of algorithm)

- **Syntax error**—This is a mistake in the program where the rules of the programming language are not followed. (missing parenthesis, incorrect indentation, misspelling name of function calls)

- **Runtime error**—This is a mistake in the program that occurs during the execution of a program. Programming languages define their own runtime errors.(divide by 0, accessing out-of-bounds index of a list)

Big Idea 1

Read the code below. Assume that myList is a nonempty list of numbers. Identify the error. What kind of error is it?

```
Line 1: Procedure getTotal(myList)
Line 2: {
Line 3:   total ← myList[1]
Line 3:   FOR EACH item IN myList
Line 4:   {
Line 5:    total ← total + item
Line 6:   }
Line 7: RETURN(total)
Line 8: }
```

Logic Error. The code adds the first number twice in the sum.

References

Reichelson, Seth. AP Computer Science Principles Premium with 6 Practice Tests (Barron's Test Prep) (p. 92). Barrons Educational Series.