

AP COMPUTER SCIENCE A

UNIT 10

Recursion



5-7.5%
AP EXAM WEIGHTING



~3-5
CLASS PERIODS

The icon consists of a white circle containing a blue square with the letters 'AP' in white. Below the square is a small blue monitor-like shape with two lines representing a stand.

Remember to go to [AP Classroom](#) to assign students the online **Personal Progress Check** for this unit.

Whether assigned as homework or completed in class, the **Personal Progress Check** provides each student with immediate feedback related to this unit's topics and skills.

Personal Progress Check 10

Multiple-choice: ~10 questions

Free-response: 1 question

- Methods and Control Structures (recursive and non-recursive solutions allowed)

Recursion



Developing Understanding

BIG IDEA 1

Control **CON**

- What real-world processes do you follow that are recursive in nature?
- Why do programmers sometimes prefer using recursive solutions when sorting data in a large data set?

Sometimes a problem can be solved by solving smaller or simpler versions of the same problem rather than attempting an iterative solution. This is called *recursion*, and it is a powerful math and computer science idea. In this unit, students will revisit how control is passed when methods are called, which is necessary knowledge when working with recursion. Tracing skills introduced in Unit 2 are helpful for determining the purpose or output of a recursive method. In this unit, students will learn how to write simple recursive methods and determine the purpose or output of a recursive method by tracing.

Building Computational Thinking Practices

1.B 2.C 2.D

To better understand how recursion works, students should spend time writing their own recursive methods. Often, this can be overwhelming for students. One way to scaffold this skill for students is to require them to write a portion of program code, such as the base case, that can be used to complete the recursive method.

Students should be able to determine the result of recursive method calls. Tracing the series of calls is a useful way to glean what the recursive method is doing and how it is accomplishing its purpose. Recursive algorithms, such as sorting and searching algorithms, often produce a result much more quickly than iterative solutions. Students also need to understand how many times statements in a recursive solution execute based on given input values.

Preparing for the AP Exam


Recursion is primarily assessed through the multiple-choice section of the exam. Students are often asked to determine the result of a specific call to a recursive method or to describe the behavior of a recursive method. A call to a recursive method is just like a call to a nonrecursive method. Because a method is an abstraction of a process that has a specific result for each varied input, using a specific input value provides an understanding of how the method functions for that input. By understanding several instances of the method call, students can abstract and generalize the method's overall purpose or process.

While students will not be required to write a recursive solution in the free-response section, recursive solutions are often a more straightforward way of writing the solutions than iterative designs. Writing recursive solutions and analyzing calls to recursive methods help engage students with all aspects of recursive methods and provide them with a deeper understanding of how recursion works.

**UNIT
10**

Recursion

UNIT AT A GLANCE

Enduring Understanding	Topic	Suggested Skills	Class Periods
			~3-5 CLASS PERIODS
CON-2	10.1 Recursion	<p>1.B Determine code that would be used to complete code segments.</p> <p>5.A Describe the behavior of a given segment of program code.</p>	
	10.2 Recursive Searching and Sorting	<p>2.C Determine the result or output based on the statement execution order in a code segment containing method calls.</p> <p>2.D Determine the number of times a code segment will execute.</p>	
<p> Go to AP Classroom to assign the Personal Progress Check for Unit 10. Review the results in class to identify and address any student misunderstandings.</p>			

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 in this unit. Please refer to the Instructional Approaches section beginning on p. 159 for more examples of activities and strategies.

Activity	Topic	Sample Activity
1	10.1	<p>Sharing and responding</p> <p>Provide students with the pseudocode to multiple recursive algorithms, and have students write the base case of the recursive methods and share it with their partner. The partner should then provide feedback, including any corrections or additions that may be needed.</p>
2	10.1	<p>Look for a pattern</p> <p>Provide students with a recursive method and several different inputs. Have students run the recursive method, record the various outputs, and look for a pattern between the input and related output. Ask students to write one or two sentences as a broad description of what the recursive method is doing.</p>
3	10.2	<p>Code tracing</p> <p>When looking at a recursive method to determine how many times it executes, have students create a call tree or a stack trace to show the method being called and the values of any parameters of each call. Students can then count up the number of times a statement executes or a method is called.</p>



Unit Planning Notes

Use the space below to plan your approach to the unit. Consider how you want to pace your course and where you will incorporate analysis of recursive program code.

.....

.....

.....

SUGGESTED SKILLS

1.B

Determine code that would be used to complete code segments.

5.A

Describe the behavior of a given segment of program code.



AVAILABLE RESOURCES

- Runestone Academy: AP CSA—Java Review: 12—Recursion
- Practice-It!: BJP4 Chapter 12: Recursion—Self-Check 12.3–12.6, 12.13–12.15
- CodingBat Java: Recursion

TOPIC 10.1

Recursion

Required Course Content

ENDURING UNDERSTANDING

CON-2

Programmers incorporate iteration and selection into code as a way of providing instructions for the computer to process each of the many possible input values.

LEARNING OBJECTIVE

CON-2.0

Determine the result of executing recursive methods.

ESSENTIAL KNOWLEDGE

CON-2.0.1

A recursive method is a method that calls itself.

CON-2.0.2

Recursive methods contain at least one base case, which halts the recursion, and at least one recursive call.

CON-2.0.3

Each recursive call has its own set of local variables, including the formal parameters.

CON-2.0.4

Parameter values capture the progress of a recursive process, much like loop control variable values capture the progress of a loop.

CON-2.0.5

Any recursive solution can be replicated through the use of an iterative approach.

✘ EXCLUSION STATEMENT—(EK CON-2.0.5):

Writing recursive program code is outside the scope of the course and AP Exam.

CON-2.0.6

Recursion can be used to traverse `String`, `array`, and `ArrayList` objects.

TOPIC 10.2

Recursive Searching and Sorting

Required Course Content

ENDURING UNDERSTANDING

CON-2

Programmers incorporate iteration and selection into code as a way of providing instructions for the computer to process each of the many possible input values.

LEARNING OBJECTIVE

CON-2.P

Apply recursive search algorithms to information in `String`, 1D array, or `ArrayList` objects.

CON-2.Q

Apply recursive algorithms to sort elements of array or `ArrayList` objects.

ESSENTIAL KNOWLEDGE

CON-2.P.1

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

CON-2.P.2

The binary search algorithm starts at the middle of a sorted array or `ArrayList` and eliminates half of the array or `ArrayList` in each iteration until the desired value is found or all elements have been eliminated.

CON-2.P.3

Binary search can be more efficient than sequential/linear search.

EXCLUSION STATEMENT—(EK CON-2.P.3): Search algorithms other than sequential/linear and binary search are outside the scope of the course and AP Exam.

CON-2.P.4

The binary search algorithm can be written either iteratively or recursively.

CON-2.Q.1

Merge sort is a recursive sorting algorithm that can be used to sort elements in an array or `ArrayList`.

SUGGESTED SKILLS

2.C

Determine the result or output based on the statement execution order in a code segment containing method calls.

2.D

Determine the number of times a code segment will execute.



AVAILABLE RESOURCES

- [Runestone Academy: AP CSA—Java Review: 13.3—Binary Search](#)
- [Runestone Academy: AP CSA—Java Review: 13.6—Merge Sort](#)
- [Practice-It!: BJP4 Chapter 12: Recursion—Exercises 12.1–12.3, 12.6–12.14, 12.18–12.22](#)
- [Practice-It!: BJP4 Chapter 13: Searching and Sorting—Self-Check 12.30](#)

