AP COMPUTER SCIENCE A

UNIT 1 Primitive Types



AP

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 1

Multiple-choice: ~25 questions



BIG IDEA 1

Modularity MOD

 How can we use programs to solve problems?

BIG IDEA 2 Variables VAR

 In what ways are numbers used in the programs and apps you use most often?

BIG IDEA 3 Control CON

 How are mathematical concepts being used in the programs and apps that you use most often?

↔ Developing Understanding

This unit introduces students to the Java programming language and the use of classes, providing students with a firm foundation of concepts that will be leveraged and built upon in all future units. Students will focus on writing the main method and will start to call preexisting methods to produce output. The use of preexisting methods for input is not prescribed in the course; however, input is a necessary part of any computer science course so teachers will need to determine how they will address this in their classrooms. Students will start to learn about three built-in data types and learn how to create variables, store values, and interact with those variables using basic operations. The ability to write expressions is essential to representing the variability of the real world in a program and will be used in all future units. Primitive data is one of two categories of variables covered in this course. The other category, reference data, will be covered in Unit 2.

Building Computational Thinking Practices

When writing programs, programmers use mathematical expressions to represent the relationships between quantities in the real world. This requires programmers to apply the meaning of specific Java operators to these formulas and expressions. While practicing writing these expressions, students should begin to understand that programs are composed of a series of statements that use arithmetic operators.

During the early stages of learning to program, have students look at existing program code, and ask them to explain what it does rather than having them develop program code from scratch.

When writing code, errors are inevitable. To learn how to identify and correct errors, students need exposure to the error messages generated by the compiler. The ability to communicate to collaborative partners why a code segment will not compile or work as intended will aid students in being able to correct the error and build working programs that accomplish specific tasks.

Preparing for the AP Exam

This unit provides a lot of the foundational content and skills that students will continue to draw on throughout the course. While much of what is covered in this unit is not explicitly assessed on the AP Exam, this content exists in the program code of nearly every assessment question. It is important for students to spend adequate time practicing writing expressions and applying meaning to specific operators during this unit. Early success will build students' confidence, which will be necessary as we build on this knowledge, adding new concepts and requiring more sophisticated application of the concepts.



UNIT AT A GLANCE

ng standing			Class Periods
Enduri Unders	Торіс	Suggested Skills	~8-10 CLASS PERIODS
MOD-1 VAR-1	1.1 Why Programming? Why Java?	 2.B Determine the result or output based on statement execution order in a code segment without method calls (other than output). 4.B Identify errors in program code. 	
VAR-1	1.2 Variables and Data Types	 Determine an appropriate program design to solve a problem or accomplish a task (<i>not</i> <i>assessed</i>). Determine code that would be used to complete code segments. 	
CON-1	1.3 Expressions and Assignment Statements	 Determine code that would be used to complete code segments. Apply the meaning of specific operators. 	
	1.4 Compound Assignment Operators	 2.B Determine the result or output based on statement execution order in a code segment without method calls (other than output). 5.A Describe the behavior of a given segment of program code. 	
	1.5 Casting and Ranges of Variables	 2.B Determine the result or output based on statement execution order in a code segment without method calls (other than output). 5.B Explain why a code segment will not compile or work as intended. 	
AP	Go to AP Classroom to assign the Review the results in class to identif	Personal Progress Check for Unit 1. y and address any student misunderstandings.	



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	1.1	Error analysis Provide students with code that contains syntax errors. Ask students to identify and correct the errors in the provided code. Once they feel they have identified and corrected all syntax errors, have them verify their conclusion by using a compiler and an IDE that does not autocorrect errors.
2	1.3	Activating prior knowledge The basic arithmetic operators of +, –, /, and * are similar to what students have experienced in math class or when using a calculator. Give students a list of expressions and ask them to apply what they know from math class to evaluate the meaning of the expressions. Have them verify their results by putting them into a compiler.
3	1.4	Sharing and responding Put student into groups of two. Provide each student with a different set of statements; in each pair, one student should have a list of statements that contain compound assignment operators, while the other student should have a list of statements that accomplish the same thing without using compound statements. Be sure the statements are in a different order. Students should take turns describing what a statement does to their partner, and the partner should determine which statement of theirs is equivalent to the one being described.
4	1.5	Predict and compare Provide students with several statements that involve casting. Each cast should be on a different value in the statement. Have students predict the resulting value. For any statements that would not compile or work as intended, have students explain the problem and propose a solution. They should verify their results by putting those results into a compiler.



SUGGESTED SKILLS

Determine the result or output based on statement execution order in a code segment without method calls (other than output).

UNIT

4.B

Identify errors in program code.

AVAILABLE RESOURCE

 Runestone Academy: AP CSA—Java Review: 2.1—What is Java?

TOPIC 1.1 Why Programming? Why Java?

Required Course Content

ENDURING UNDERSTANDING

MOD-1

Some objects or concepts are so frequently represented that programmers can draw upon existing code that has already been tested, enabling them to write solutions more quickly and with a greater degree of confidence.

LEARNING OBJECTIVE

MOD-1.A

Call System class methods to generate output to the console.

ESSENTIAL KNOWLEDGE

MOD-1.A.1

System.out.print and System.out. println display information on the computer monitor.

MOD-1.A.2

System.out.println moves the cursor to a new line after the information has been displayed, while System.out.print does not.

ENDURING UNDERSTANDING

VAR-1

To find specific solutions to generalizable problems, programmers include variables in their code so that the same algorithm runs using different input values.

LEARNING OBJECTIVE

ESSENTIAL KNOWLEDGE

VAR-1.A Create string literals.

VAR-1.A.1 A string literal is enclosed in double quotes.

TOPIC 1.2 Variables and Data Types

Required Course Content

ENDURING UNDERSTANDING

VAR-1

To find specific solutions to generalizable problems, programmers include variables in their code so that the same algorithm runs using different input values.

LEARNING OBJECTIVE

VAR-1.B

Identify the most appropriate data type category for a particular specification.

VAR-1.C

Declare variables of the correct types to represent primitive data.

ESSENTIAL KNOWLEDGE

VAR-1.B.1

A type is a set of values (a domain) and a set of operations on them.

VAR-1.B.2

Data types can be categorized as either primitive or reference.

VAR-1.B.3

The primitive data types used in this course define the set of operations for numbers and Boolean values.

VAR-1.C.1

The three primitive data types used in this course are int, double, and boolean.

VAR-1.C.2

Each variable has associated memory that is used to hold its value.

VAR-1.C.3

The memory associated with a variable of a primitive type holds an actual primitive value.

VAR-1.C.4

When a variable is declared final, its value cannot be changed once it is initialized.

SUGGESTED SKILLS

Determine an appropriate program design to solve a problem or accomplish a task.

UNIT

1.B

Determine code that would be used to complete code segments.



AVAILABLE RESOURCE Runestone Academy:

AP CSA—Java Review: 3—Variables



SUGGESTED SKILLS

Determine code that would be used to complete code segments.

2.A Apply the meaning of specific operators.



AVAILABLE RESOURCES

- Runestone Academy: AP CSA—Java Review: 3.5—Operators
- Problets: Arithmetic Expressions in Java

TOPIC 1.3 Expressions and Assignment Statements

Required Course Content

ENDURING UNDERSTANDING

CON-1

The way variables and operators are sequenced and combined in an expression determines the computed result.

LEARNING OBJECTIVE

CON-1.A

Evaluate arithmetic expressions in a program code.

ESSENTIAL KNOWLEDGE

CON-1.A.1

A literal is the source code representation of a fixed value.

CON-1.A.2

Arithmetic expressions include expressions of type int and double.

CON-1.A.3

The arithmetic operators consist of +, –, *, /, and .

CON-1.A.4

An arithmetic operation that uses two int values will evaluate to an int value.

CON-1.A.5

An arithmetic operation that uses a double value will evaluate to a double value.

CON-1.A.6

Operators can be used to construct compound expressions.

CON-1.A.7

During evaluation, operands are associated with operators according to operator precedence to determine how they are grouped.

CON-1.A.8

An attempt to divide an integer by zero will result in an ArithmeticException to occur.

continued on next page



LEARNING OBJECTIVE

CON-1.B

Evaluate what is stored in a variable as a result of an expression with an assignment statement.

ESSENTIAL KNOWLEDGE

CON-1.B.1

The assignment operator (=) allows a program to initialize or change the value stored in a variable. The value of the expression on the right is stored in the variable on the left.

CON-1.B.2

During execution, expressions are evaluated to produce a single value.

CON-1.B.3

The value of an expression has a type based on the evaluation of the expression.



SUGGESTED SKILLS

Determine the result or output based on statement execution order in a code segment without method calls (other than output).

UNIT

5.A

Describe the behavior of a given segment of program code.

AVAILABLE RESOURCE

 Runestone Academy: AP CSA—Java Review: 3.5—Operators

TOPIC 1.4 Compound Assignment Operators

Required Course Content

ENDURING UNDERSTANDING

CON-1

The way variables and operators are sequenced and combined in an expression determines the computed result.

LEARNING OBJECTIVE

CON-1.B

Evaluate what is stored in a variable as a result of an expression with an assignment statement.

ESSENTIAL KNOWLEDGE

CON-1.B.4

Compound assignment operators (+=, -=, *=, /=, \$=) can be used in place of the assignment operator.

CON-1.B.5

The increment operator (++) and decrement operator (--) are used to add 1 or subtract 1 from the stored value of a variable or an array element. The new value is assigned to the variable or array element.

EXCLUSION STATEMENT—(EK CON-1.B.5): The use of increment and decrement operators in prefix form (i.e., ++x) and inside other expressions (i.e., arr[x++]) is outside the scope of this course and the AP Exam.

TOPIC 1.5 Casting and Ranges of Variables

Required Course Content

ENDURING UNDERSTANDING

CON-1

The way variables and operators are sequenced and combined in an expression determines the computed result.

LEARNING OBJECTIVE

CON-1.C

Evaluate arithmetic expressions that use casting.

ESSENTIAL KNOWLEDGE

CON-1.C.1

The casting operators (int) and (double) can be used to create a temporary value converted to a different data type.

CON-1.C.2

Casting a double value to an int causes the digits to the right of the decimal point to be truncated.

CON-1.C.3

Some programming code causes int values to be automatically cast (widened) to double values.

CON-1.C.4

Values of type double can be rounded to the nearest integer by (int) (x + 0.5) or (int) (x - 0.5) for negative numbers.

CON-1.C.5

Integer values in Java are represented by values of type int, which are stored using a finite amount (4 bytes) of memory. Therefore, an int value must be in the range from Integer.MIN_VALUE to Integer.MAX_ VALUE inclusive.

CON-1.C.6

If an expression would evaluate to an int value outside of the allowed range, an integer overflow occurs. This could result in an incorrect value within the allowed range.

SUGGESTED SKILLS

Determine the result or output based on statement execution order in a code segment without method calls (other than output).

UNIT

5.B

Explain why a code segment will not compile or work as intended.

AVAILABLE RESOURCE

 Runestone Academy: AP CSA—Java Review: 3.6—Casting

AP COMPUTER SCIENCE A

UNIT 2 Using Objects



AP

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 2

Multiple-choice: ~25 questions Free-response: 1 question

 Methods and Control Structures: partial

BIG IDEA 1

Modularity MOD

 How can we simulate election results using existing program code?

BIG IDEA 2 Variables VAR

 How are appropriate variables chosen to represent a remote control?

BIG IDEA 3 Control CON

 How do the games we play simulate randomness?

Developing Understanding

In the first unit, students used primitive types to represent real-world data and determined how to use them in arithmetic expressions to solve problems. This unit introduces a new type of data: reference data. Reference data allows real-world objects to be represented in varying degrees specific to a programmer's purpose. This unit builds on students' ability to write expressions by introducing them to Math class methods to write expressions for generating random numbers and other more complex operations. In addition, strings and the existing methods within the String class are an important topic within this unit. Knowing how to declare variables or call methods on objects is necessary throughout the course but will be very important in Units 5 and 9 when teaching students how to write their own classes and about inheritance relationships.

Building Computational Thinking Practices

The study of computer science involves implementing the design or specification for a program. This is the fun and rewarding part of computer science, because it involves putting a plan into practice to create a runnable program. In addition to developing their own programs, students should practice completing partially written program code to fulfill a specification. This builds their confidence and provides them opportunities to be successful during these early stages of learning.

Programmers often rely on existing program code to build new programs. Using existing code saves time, because it has already been tested. By using the String class, students will learn how to interact with and utilize any existing Java class to create objects and call methods.

Preparing for the AP Exam

During the free-response portion of the exam, students will be required to call methods of classes that they haven't been exposed to prior to the exam. Students should get plenty of practice identifying the proper parameters to use when calling methods of classes that are provided to them.

Often, students struggle with free-response questions that require them to work with the String class. Using the Java Quick Reference (p. 209) regularly during class will help students become more familiar with this resource prior to the exam. Paying close attention to the method descriptions will ensure that students use the correct type and order of parameters when calling String methods.

Practice close reading techniques with students prior to the exam, such as underlining keywords, return types, and parameters. Students have approximately 20 minutes to read, process, and answer each of the four free-response questions. These close reading techniques are valuable in helping students process the questions quickly without inadvertently missing key information.



UNIT AT A GLANCE

ing standing			Class Periods
Enduri Under	Торіс	Suggested Skills	~13-15 CLASS PERIODS
MOD-1	2.1 Objects: Instances of Classes	5.A Describe the behavior of a given segment of program code.	
MOD-1 VAR-1	2.2 Creating and Storing Objects (Instantiation)	1.C Determine code that would be used to interact with completed program code.	
		3.A Write program code to create objects of a class and call methods.	
	2.3 Calling a Void Method	1.C Determine code that would be used to interact with completed program code.	
		3.A Write program code to create objects of a class and call methods.	
	2.4 Calling a Void Method with	2. Determine the result or output based on the statement execution order in a code segment containing method calls.	
MO	Parameters	3.A Write program code to create objects of a class and call methods.	
	2.5 Calling a Non-void Method	1.C Determine code that would be used to interact with completed program code.	
		3.A Write program code to create objects of a class and call methods.	
VAR-1	2.6 String Objects: Concatenation, Literals, and More	2.A Apply the meaning of specific operators.	
	2.7 String Methods	2.C Determine the result or output based on the statement execution order in a code segment containing method calls.	
		3.A Write program code to create objects of a class and call methods.	
	2.8 Wrapper Classes: Integer and Double	2.C Determine the result or output based on the statement execution order in a code segment containing method calls.	
1-1-NC	2.9 Using the Math Class	1.B Determine code that would be used to complete code segments.	
N N N		3.A Write program code to create objects of a class and call methods.	

AP

Go to **AP Classroom** to assign the **Personal Progress Check** for Unit 2. Review the results in class to identify and address any student misunderstandings.



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	2.1	Using manipulatives When introducing students to the idea of creating objects, you can use a cookie cutter and modeling clay or dough, with the cutter representing the class and the cut dough representing the objects. For each object cut, write the instantiation. Ask students to describe what the code is doing and how the different parameter values (e.g., thickness, color) change the object that was created.
2	2.2	Marking the text Provide students with several statements that define a variable and create an object on a single line. Have students mark up the statements by circling the assignment operator and the new keyword. Then, have students underline the variable type and the constructor. Lastly, have them draw a rectangle around the list of actual parameters being passed to the constructor. Using these marked-up statements, ask students to create several new variables and objects.
3	2.9	Think-pair-share Provide students with several code segments, each with a missing expression that would contain a call to a method in the Math class, and a description of the intended outcome of each code segment. Ask them which statement should be used to complete the code segment. Have them share their responses with a partner to compare answers and come to agreement, and then have groups share with the entire class.

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 writing and analyzing program code.



SUGGESTED SKILL

5.A Describe the behavior of a given segment of program code.

UNIT



AVAILABLE RESOURCE Runestone Academy: AP CSA—Java Review: 2.2—What is a Class and an Object?

TOPIC 2.1 Objects: Instances of Classes

Required Course Content

ENDURING UNDERSTANDING

MOD-1

Some objects or concepts are so frequently represented that programmers can draw upon existing code that has already been tested, enabling them to write solutions more quickly and with a greater degree of confidence.

LEARNING OBJECTIVE

MOD-1.B

Explain the relationship between a class and an object.

ESSENTIAL KNOWLEDGE

MOD-1.B.1

An object is a specific instance of a class with defined attributes.

MOD-1.B.2

A class is the formal implementation, or blueprint, of the attributes and behaviors of an object.

TOPIC 2.2 Creating and Storing Objects (Instantiation)

Required Course Content

ENDURING UNDERSTANDING

MOD-1

Some objects or concepts are so frequently represented that programmers can draw upon existing code that has already been tested, enabling them to write solutions more quickly and with a greater degree of confidence.

LEARNING OBJECTIVE

MOD-1.C

Identify, using its signature, the correct constructor being called.

ESSENTIAL KNOWLEDGE

MOD-1.C.1

A signature consists of the constructor name and the parameter list.

MOD-1.C.2

The parameter list, in the header of a constructor, lists the types of the values that are passed and their variable names. These are often referred to as formal parameters.

MOD-1.C.3

A parameter is a value that is passed into a constructor. These are often referred to as actual parameters.

MOD-1.C.4

Constructors are said to be overloaded when there are multiple constructors with the same name but a different signature.

MOD-1.C.5

The actual parameters passed to a constructor must be compatible with the types identified in the formal parameter list.

MOD-1.C.6

Parameters are passed using call by value. Call by value initializes the formal parameters with copies of the actual parameters.

continued on next page

SUGGESTED SKILLS

1.C

Determine code that would be used to interact with completed program code.

UNIT

3.A

Write program code to create objects of a class and call methods.

LEARNING OBJECTIVE

MOD-1.D

For creating objects:

- a. Create objects by calling constructors without parameters.
- b. Create objects by calling constructors with parameters.

ESSENTIAL KNOWLEDGE

MOD-1.D.1

Every object is created using the keyword new followed by a call to one of the class's constructors.

MOD-1.D.2

A class contains constructors that are invoked to create objects. They have the same name as the class.

MOD-1.D.3

Existing classes and class libraries can be utilized as appropriate to create objects.

MOD-1.D.4

Parameters allow values to be passed to the constructor to establish the initial state of the object.

ENDURING UNDERSTANDING

VAR-1

To find specific solutions to generalizable problems, programmers include variables in their code so that the same algorithm runs using different input values.

LEARNING OBJECTIVE

VAR-1.D

Define variables of the correct types to represent reference data.

ESSENTIAL KNOWLEDGE

VAR-1.D.1

The keyword null is a special value used to indicate that a reference is not associated with any object.

VAR-1.D.2

The memory associated with a variable of a reference type holds an object reference value or, if there is no object, null. This value is the memory address of the referenced object.

TOPIC 2.3 Calling a Void Method

Required Course Content

ENDURING UNDERSTANDING

MOD-1

Some objects or concepts are so frequently represented that programmers can draw upon existing code that has already been tested, enabling them to write solutions more quickly and with a greater degree of confidence.

LEARNING OBJECTIVE

MOD-1.E

Call non-static void methods without parameters.

ESSENTIAL KNOWLEDGE

MOD-1.E.1

An object's behavior refers to what the object can do (or what can be done to it) and is defined by methods.

MOD-1.E.2

Procedural abstraction allows a programmer to use a method by knowing what the method does even if they do not know how the method was written.

MOD-1.E.3

A method signature for a method without parameters consists of the method name and an empty parameter list.

MOD-1.E.4

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

continued on next page



1.C

Determine code that would be used to interact with completed program code.

UNIT



Write program code to create objects of a class and call methods.



AVAILABLE RESOURCE

 Classroom Resources > GridWorld Case Study: Part I

LEARNING OBJECTIVE

MOD-1.E Call non-static void methods without parameters.

ESSENTIAL KNOWLEDGE

MOD-1.E.5

Non-static methods are called through objects of the class.

MOD-1.E.6

The dot operator is used along with the object name to call non-static methods.

MOD-1.E.7

Void methods do not have return values and are therefore not called as part of an expression.

MOD-1.E.8

Using a null reference to call a method or access an instance variable causes a NullPointerException to be thrown.

TOPIC 2.4 Calling a Void Method with Parameters

Required Course Content

ENDURING UNDERSTANDING

MOD-1

Some objects or concepts are so frequently represented that programmers can draw upon existing code that has already been tested, enabling them to write solutions more quickly and with a greater degree of confidence.

LEARNING OBJECTIVE

MOD-1.F

Call non-static void methods with parameters.

ESSENTIAL KNOWLEDGE

MOD-1.F.1

A method signature for a method with parameters consists of the method name and the ordered list of parameter types.

MOD-1.F.2

Values provided in the parameter list need to correspond to the order and type in the method signature.

MOD-1.F.3

Methods are said to be overloaded when there are multiple methods with the same name but a different signature.

SUGGESTED SKILLS

UNIT

2.C

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

3.A

Write program code to create objects of a class and call methods.



AVAILABLE RESOURCE

 Practice-It!: BJP4 Chapter 3: Parameters and Objects—Self-Check 3.2–3.9



SUGGESTED SKILLS

1.C

Determine code that would be used to interact with completed program code.

UNIT

3.A

Write program code to create objects of a class and call methods.



AVAILABLE RESOURCES

 The Exam > 2018 AP Computer Science A Exam Free-Response Question #1 (Frog Simulation)

TOPIC 2.5 Calling a Non-void Method

Required Course Content

ENDURING UNDERSTANDING

MOD-1

Some objects or concepts are so frequently represented that programmers can draw upon existing code that has already been tested, enabling them to write solutions more quickly and with a greater degree of confidence.

LEARNING OBJECTIVE

MOD-1.G

Call non-static non-void methods with or without parameters.

ESSENTIAL KNOWLEDGE

MOD-1.G.1

Non-void methods return a value that is the same type as the return type in the signature. To use the return value when calling a non-void method, it must be stored in a variable or used as part of an expression.

TOPIC 2.6 String Objects: Concatenation, Literals, and More

Required Course Content

ENDURING UNDERSTANDING

VAR-1

To find specific solutions to generalizable problems, programmers include variables in their code so that the same algorithm runs using different input values.

LEARNING OBJECTIVE

VAR-1.E

For String class: a. Create String objects. b. Call String methods.

ESSENTIAL KNOWLEDGE

VAR-1.E.1

String objects can be created by using string literals or by calling the String class constructor.

VAR-1.E.2

String objects are immutable, meaning that String methods do not change the String object.

VAR-1.E.3

String objects can be concatenated
using the + or += operator, resulting in a new
String object.

VAR-1.E.4

Primitive values can be concatenated with a String object. This causes implicit conversion of the values to String objects.

VAR-1.E.5

Escape sequences start with a $\$ and have a special meaning in Java. Escape sequences used in this course include $\$ ", $\$, and $\$ n.

specific operators.	

UNIT

SUGGESTED SKILL

2.A

AVAILABLE RESOURCES

- Runestone Academy: AP CSA—Java Review: 4—Strings
- Practice-It!: BJP4 Chapter 3: Parameters and Objects—Self-Check 3.18



SUGGESTED SKILLS

2.C

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

3.A

Write program code to create objects of a class and call methods.



AVAILABLE RESOURCES

- Java Quick Reference (see Appendix)
- Runestone Academy: AP CSA—Java Review: 4.3—String Methods on the Exam
- CodingBat Java: String-1
- Practice-It!: BJP4 Chapter 3: Parameters and Objects—Self-Check 3.19 and 3.20

TOPIC 2.7 String Methods

Required Course Content

ENDURING UNDERSTANDING

VAR-1

To find specific solutions to generalizable problems, programmers include variables in their code so that the same algorithm runs using different input values.

LEARNING OBJECTIVE

VAR-1.E For String class: a. Create String objects. b. Call String methods.

ESSENTIAL KNOWLEDGE

VAR-1.E.6

Application program interfaces (APIs) and libraries simplify complex programming tasks.

VAR-1.E.7

Documentation for APIs and libraries are essential to understanding the attributes and behaviors of an object of a class.

VAR-1.E.8

Classes in the APIs and libraries are grouped into packages.

VAR-1.E.9

The String class is part of the java.lang package. Classes in the java.lang package are available by default.

VAR-1.E.10

A String object has index values from 0 to length - 1. Attempting to access indices outside this range will result in an IndexOutOfBoundsException.

VAR-1.E.11

A String object can be concatenated with an object reference, which implicitly calls the referenced object's toString method.

continued on next page

LEARNING OBJECTIVE

VAR-1.E For String class: a. Create String objects. b. Call String methods.

ESSENTIAL KNOWLEDGE

VAR-1.E.12

The following String methods and constructors—including what they do and when they are used—are part of the Java Quick Reference:

- String(String str) Constructs a new String object that represents the same sequence of characters as str
- int length() Returns the number of characters in a String object
- String substring(int from, int to) — Returns the substring beginning at index from and ending at index to - 1
- String substring(int from) — Returns substring(from, length())
- int indexOf(String str) Returns the index of the first occurrence of str; returns -1 if not found
- boolean equals(String other)

 Returns true if this is equal to
 other; returns false otherwise
- int compareTo(String other)

 Returns a value < 0 if this is less
 than other; returns zero if this is equal
 to other; returns a value > 0 if this is
 greater than other

VAR-1.E.13

A string identical to the single element substring at position index can be created by calling substring(index, index + 1).



SUGGESTED SKILL

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



AVAILABLE RESOURCE

 Java Quick Reference (see Appendix)

TOPIC 2.8 Wrapper Classes: Integer and Double

Required Course Content

ENDURING UNDERSTANDING

VAR-1

To find specific solutions to generalizable problems, programmers include variables in their code so that the same algorithm runs using different input values.

LEARNING OBJECTIVE

VAR-1.F

For wrapper classes:

- a. Create Integer objects.
- b. Call Integer methods.
- c. Create Double objects.
- d. Call Double methods.

ESSENTIAL KNOWLEDGE

VAR-1.F.1

The Integer class and Double class are part of the java.lang package.

VAR-1.F.2

The following Integer methods and constructors—including what they do and when they are used—are part of the Java Quick Reference:

- Integer(int value) Constructs a new Integer object that represents the specified int value
- Integer.MIN_VALUE The minimum value represented by an int or Integer
- Integer.MAX_VALUE The maximum value represented by an int or Integer
- int intValue() Returns the value of this Integer as an int

continued on next page

LEARNING OBJECTIVE

VAR-1.E

For wrapper classes:

- a. Create Integer objects.
- b. Call Integer methods.
- c. Create Double objects.
- d. Call Double methods.

ESSENTIAL KNOWLEDGE

VAR-1.F.3

The following Double methods and constructors—including what they do and when they are used—are part of the Java Quick Reference:

- Double(double value) Constructs a new Double object that represents the specified double value
- double doubleValue() Returns the value of this Double as a double

VAR-1.F.4

Autoboxing is the automatic conversion that the Java compiler makes between primitive types and their corresponding object wrapper classes. This includes converting an int to an Integer and a double to a Double.

VAR-1.F.5

The Java compiler applies autoboxing when a primitive value is:

- Passed as a parameter to a method that expects an object of the corresponding wrapper class.
- Assigned to a variable of the corresponding wrapper class.

VAR-1.F.6

Unboxing is the automatic conversion that the Java compiler makes from the wrapper class to the primitive type. This includes converting an Integer to an int and a Double to a double.

VAR-1.F.7

The Java compiler applies unboxing when a wrapper class object is:

- Passed as a parameter to a method that expects a value of the corresponding primitive type.
- Assigned to a variable of the corresponding primitive type.



SUGGESTED SKILLS

1.B

Determine code that would be used to complete code segments.

3.A

Write program code to create objects of a class and call methods.



AVAILABLE RESOURCES

- Java Quick Reference (see Appendix)
- Practice-It!: BJP4 Chapter 3: Parameters and Objects— Exercises 3.7 and 3.8

TOPIC 2.9 Using the Math Class

Required Course Content

ENDURING UNDERSTANDING

MOD-1

MOD-1.H

Call static methods.

Some objects or concepts are so frequently represented that programmers can draw upon existing code that has already been tested, enabling them to write solutions more quickly and with a greater degree of confidence.

MOD-1.H.1

LEARNING OBJECTIVE

ESSENTIAL KNOWLEDGE

Static methods are called using the dot operator along with the class name unless they are defined in the enclosing class.

ENDURING UNDERSTANDING

CON-1

The way variables and operators are sequenced and combined in an expression determines the computed result.

LEARNING OBJECTIVE

CON-1.D

Evaluate expressions that use the Math class methods.

ESSENTIAL KNOWLEDGE

CON-1.D.1

The Math class is part of the java.lang package.

CON-1.D.2

The Math class contains only static methods.

continued on next page

LEARNING OBJECTIVE

CON-1.D

Evaluate expressions that use the Math class methods.

ESSENTIAL KNOWLEDGE

CON-1.D.3

The following static Math methods—including what they do and when they are used—are part of the Java Quick Reference:

- int abs(int x) Returns the absolute value of an int value
- double abs(double x) Returns the absolute value of a double value
- double pow(double base, double exponent) — Returns the value of the first parameter raised to the power of the second parameter
- double sqrt(double x) Returns the positive square root of a double value
- double random() Returns a double value greater than or equal to 0.0 and less than 1.0

CON-1.D.4

The values returned from Math.random can be manipulated to produce a random int or double in a defined range.

AP COMPUTER SCIENCE A

UNIT 3 Boolean Expressions and if Statements



AP

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 3

Multiple-choice: ~20 questions Free-response: 2 questions

- Methods and Control Structures
- Methods and Control Structures: partial

Boolean Expressions and if Statements

BIG IDEA 1 Control CON

- How can you use different conditional statements to write a pick-your-own-path interactive story?
- Why is selection a necessary part of programming languages?

Developing Understanding

UNIT

3

Algorithms are composed of three building blocks: sequencing, selection, and iteration. This unit focuses on selection, which is represented in a program by using conditional statements. Conditional statements give the program the ability to decide and respond appropriately and are a critical aspect of any nontrivial computer program. In addition to learning the syntax and proper use of conditional statements, students will build on the introduction of Boolean variables by writing Boolean expressions with relational and logical operators.

The third building block of all algorithms is iteration, which you will cover in Unit 4. Selection and iteration work together to solve problems.

Building Computational Thinking Practices 2.B 3.C 4.A 4.C

Selection allows programmers to incorporate choice into their programs: to create games that react to interactions of the user, to develop simulations that are more real world by allowing for variability, or to discover new knowledge in a sea of information by filtering out irrelevant data. Students should be able to write program code that uses conditional statements for selection. Have students write their program code on paper and trace it with different sample inputs before writing code on the computer.

Programmers need to make design decisions when creating programs that determine how a program specification will be implemented. Typically, there are many ways in which statements can be written to yield the same result, and this final determination is dictated by the programmer. Exposing students to many different code segments that yield equivalent results allows them to be more confident in their solution and helps expose them to new ways of solving the problem that may be better than their current solution.

Preparing for the AP Exam

The study of computer science involves the analysis of program code. On the multiplechoice section of the exam, students will be asked to determine the result of a given program code segment based on specific input and on the behavior of program code in general. Students should be able to determine the result of program code that uses conditional statements and nested conditional statements to represent nonlinear processes in a program.

Often, students will write program code that mishandles one of the given conditions. The ability to trace program code can be valuable when testing programs to ensure that all conditions are met. Testing for the different expected behaviors of conditional statements is a critical part of program development and is useful when writing program code or analyzing given code segments. Students should develop sample test cases to illustrate each unique behavior to aid in finding errors and validating results.



UNIT AT A GLANCE

uring erstanding			Class Periods
End	Торіс	Suggested Skills	~11-13 CLASS PERIODS
CON-1	3.1 Boolean Expressions	2.A Apply the meaning of specific operators.	
CON-2	3.2 if Statements and Control Flow	2.3 Determine the result or output based on statement execution order in a code segment without method calls (other than output).	
		SC Write program code to satisfy method specifications using expressions, conditional statements, and iterative statements.	
	3.3 if-else Statements	SC Write program code to satisfy method specifications using expressions, conditional statements, and iterative statements.	
		4.A Use test-cases to find errors or validate results.	
	3.4 else if Statements	SC Write program code to satisfy method specifications using expressions, conditional statements, and iterative statements.	
		4.C Determine if two or more code segments yield equivalent results.	
CON-1, CON-2	3.5 Compound Boolean Expressions	2.B Determine the result or output based on statement execution order in a code segment without method calls (other than output).	
		BC Write program code to satisfy method specifications using expressions, conditional statements, and iterative statements.	
CON-1	3.6 Equivalent Boolean Expressions	4.C Determine if two or more code segments yield equivalent results.	
	3.7 Comparing Objects	2.C Determine the result or output based on the statement execution order in a code segment containing method calls.	
		3.A Write program code to create objects of a class and call methods.	

Go to **AP Classroom** to assign the **Personal Progress Check** for Unit 3.

AP Review the results in class to identify and address any student misunderstandings.
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	3.3	Code tracing Provide students with several code segments that contain conditional statements. Have students trace various sample inputs, keeping track of the statements that get executed and the order in which they are executed. This can help students find errors and validate results.
2	3.2–3.5	Pair programming Have students work with a partner to create a "guess checker" that could be used as part of a larger game. Students compare four given digits to a preexisting four-digit code that is stored in individual variables. Their program should provide output containing the number of correct digits in correct locations, as well as the number of correct digits in incorrect locations. This program can be continually improved as students learn about nested conditional statements and compound Boolean expressions.
3	3.5	Diagramming Have students create truth tables by listing all the possible true and false combinations and corresponding Boolean values for a given compound Boolean expression. Once students have created the truth table, provide students with input values. Have students determine the value of each individual Boolean expression and use the truth table to determine the result of the compound Boolean expression.
4	3.6	Student response system Provide students with a code segment that utilizes conditional statements and a compound Boolean expression, and ask them to choose an equivalent code segment that uses a nested conditional statement (and vice versa). Have them report their responses using a student response system.
5	3.7	Predict and compare Have students predict the output of several different code segments that compare object references—some that use .equals() and some that use ==. Once done, have them create a program that contains those code segments and compare the actual and expected results. This is best illustrated using a simple class that you write yourself.



SUGGESTED SKILL 2.A

Apply the meaning of specific operators.

Ξ

TOPIC 3.1 Boolean Expressions

Required Course Content

ENDURING UNDERSTANDING

CON-1

The way variables and operators are sequenced and combined in an expression determines the computed result.

LEARNING OBJECTIVE

CON-1.E

Evaluate Boolean expressions that use relational operators in program code.

ESSENTIAL KNOWLEDGE

CON-1.E.1

Primitive values and reference values can be compared using relational operators (i.e., == and !=).

CON-1.E.2

Arithmetic expression values can be compared using relational operators (i.e., <, >, <=, >=).

CON-1.E.3

An expression involving relational operators evaluates to a Boolean value.

AVAILABLE RESOURCES Practice-It!: BJP4 Chapter 4: Conditional Execution—Self-Check 4.2

Boolean Expressions and if Statements

торіс 3.2 if Statements and Control Flow

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.A

Represent branching logical processes by using conditional statements.

ESSENTIAL KNOWLEDGE

CON-2.A.1

Conditional statements interrupt the sequential execution of statements.

CON-2.A.2

if statements affect the flow of control by executing different statements based on the value of a Boolean expression.

CON-2.A.3

A one-way selection (if statement) is written when there is a set of statements to execute under a certain condition. In this case, the body is executed only when the Boolean condition is true.

SUGGESTED SKILLS

Determine the result or output based on statement execution order in a code segment without method calls (other than output).

UNIT

3

3.C

Write program code to satisfy method specifications using expressions, conditional statements, and iterative statements.



AVAILABLE RESOURCES

- Runestone Academy: AP CSA—Java Review: 5.1—Conditionals
- Practice-It!: BJP4 Chapter 4: Conditional Execution—Self-Check 4.3; Exercises 4.2 and 4.3
- The Exam > 2017 AP Computer Science A Exam Free-Response Question #1, Part A (Phrase)



SUGGESTED SKILLS

Write program code to satisfy method specifications using expressions, conditional statements, and iterative statements.

4.A

Use test-cases to find errors or validate results.



AVAILABLE RESOURCES

- Runestone Academy: AP CSA—Java Review: 5.1—Conditionals
- Practice-It!: BJP4 Chapter 4: Conditional Execution—Self-Check 4.5–4.6, 4.10–4.12

торіс з.з if-else Statements

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.A

Represent branching logical processes by using conditional statements.

ESSENTIAL KNOWLEDGE

CON-2.A.4

A two-way selection is written when there are two sets of statements— one to be executed when the Boolean condition is true, and another set for when the Boolean condition is false. In this case, the body of the "if" is executed when the Boolean condition is true, and the body of the "else" is executed when the Boolean condition is false.

Boolean Expressions and if Statements

торіс 3.4 else if Statements

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.A

Represent branching logical processes by using conditional statements.

ESSENTIAL KNOWLEDGE

CON-2.A.5

A multi-way selection is written when there are a series of conditions with different statements for each condition. Multi-way selection is performed using if-else-if statements such that exactly one section of code is executed based on the first condition that evaluates to true.

SUGGESTED SKILLS

UNIT

3

Write program code to satisfy method specifications using expressions, conditional statements, and iterative statements.

4.C

Determine if two or more code segments yield equivalent results.



AVAILABLE RESOURCES Runestone Academy: AP CSA—Java Review: 5.2—Three or More Options

SUGGESTED SKILLS 2.B

Determine the result or output based on statement execution order in a code segment without method calls (other than output).

3.C

Write program code to satisfy method specifications using expressions, conditional statements, and iterative statements.

AVAILABLE RESOURCES

- Runestone Academy: AP CSA—Java Review: 5.3—Complex Conditionals
- Practice-It!: BJP4 Chapter 4: Conditional Execution—Exercise 4.12

TOPIC 3.5 Compound Boolean Expressions

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

Represent branching logical

processes by using nested conditional statements.

CON-2.B

ESSENTIAL KNOWLEDGE

CON-2.B.1

Nested if statements consist of if statements within if statements.

ENDURING UNDERSTANDING

CON-1

The way variables and operators are sequenced and combined in an expression determines the computed result.

LEARNING OBJECTIVE

CON-1.F

Evaluate compound Boolean expressions in program code.

ESSENTIAL KNOWLEDGE

CON-1.F.1

Logical operators ! (not), & (and), and | (or) are used with Boolean values. This represents the order these operators will be evaluated.

CON-1.F.2

An expression involving logical operators evaluates to a Boolean value.

CON-1.F.3

When the result of a logical expression using && or | | can be determined by evaluating only the first Boolean operand, the second is not evaluated. This is known as short-circuited evaluation.

Boolean Expressions and if Statements

TOPIC 3.6 Equivalent Boolean Expressions

Required Course Content

ENDURING UNDERSTANDING

CON-1

The way variables and operators are sequenced and combined in an expression determines the computed result.

LEARNING OBJECTIVE

CON-1.G

Compare and contrast equivalent Boolean expressions.

ESSENTIAL KNOWLEDGE

CON-1.G.1

De Morgan's Laws can be applied to Boolean expressions.

CON-1.G.2

Truth tables can be used to prove Boolean identities.

CON-1.G.3

Equivalent Boolean expressions will evaluate to the same value in all cases.

SUGGESTED SKILL

Determine if two or more code segments yield equivalent results.

UNIT

3

AVAILABLE RESOURCE

Runestone Academy:

ΞÌ

AP CSA—Java Review: 5.5—De Morgan's Laws

SUGGESTED SKILLS

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

3.A

Write program code to create objects of a class and call methods.

TOPIC 3.7 Comparing Objects

Required Course Content

ENDURING UNDERSTANDING

CON-1

The way variables and operators are sequenced and combined in an expression determines the computed result.

LEARNING OBJECTIVE

CON-1.H

Compare object references using Boolean expressions in program code.

ESSENTIAL KNOWLEDGE

CON-1.H.1

Two object references are considered aliases when they both reference the same object.

CON-1.H.2

Object reference values can be compared, using == and !=, to identify aliases.

CON-1.H.3

A reference value can be compared with null, using == or !=, to determine if the reference actually references an object.

CON-1.H.4

Often classes have their own equals method, which can be used to determine whether two objects of the class are equivalent.

AP COMPUTER SCIENCE A

UNIT 4 Iteration



AP

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 4

Multiple-choice: ~15 questions Free-response: 2 questions

- Methods and Control Structures
- Methods and Control Structures: partial



17.5-22.5[%] AP EXAM WEIGHTING

~14-16 CLASS PERIODS

Iteration

BIG IDEA 1

Control CON

- How does iteration improve programs and reduce the amount of program code necessary to complete a task?
- What situations would warrant the use of one type of loop over another?

↔ Developing Understanding

This unit focuses on iteration using while and for loops. As you saw in Unit 3, Boolean expressions are useful when a program needs to perform different operations under different conditions. Boolean expressions are also one of the main components in iteration. This unit introduces several standard algorithms that use iteration. Knowledge of standard algorithms makes solving similar problems easier, as algorithms can be modified or combined to suit new situations.

Iteration is used when traversing data structures such as arrays, ArrayLists, and 2D arrays. In addition, it is a necessary component of several standard algorithms, including searching and sorting, which will be covered in later units.

Building Computational Thinking Practices

Students should be able to determine the result of program code that uses iterative statements to represent nonlinear processes in a program. Students should practice determining the number of times a given loop structure will execute. Spending time analyzing existing program code provides opportunities for students to better understand how iterative structures can be set up and used to solve their own problems, as well as the implications associated with code changes, such as how using a different iterative structure might change the result of a set of program code.

Students should be able to implement program code that uses iterative statements to represent nonlinear processes. Understanding how to write program code that repeats allows students to write programs to solve a wider variety of problems, including those that use data, which will be covered in Units 6, 7, and 8.

Preparing for the AP Exam

While the concept of iteration is tested in isolation on the multiple-choice exam, it is a foundational concept that students will use along with other topics, such as data structures, on free-response questions. Often, students struggle in situations that warrant variation in the Boolean condition of loops, such as when they want to terminate a loop early. Early termination of a loop requires the use of conditional statements within the body of the loop. If the order of the program statements is incorrect, the early termination may be triggered too early or not at all. Provide students with practice ordering statements by giving them strips of paper, each with a line of program code that can be used to assemble the correct and incorrect solutions. Ask them to reassemble the code and trace it to see if it is correct. Using manipulatives in this way makes it easier for students to rearrange the order of the program code to determine if it is in the correct order.



UNIT AT A GLANCE

'ing rstanding		Class Periods	
Endur Undei	Торіс	Suggested Skills	~14-16 CLASS PERIODS
	4.1 while Loops	1.B Determine code that would be used to complete code segments.	
		2.B Determine the result or output based on statement execution order in a code segment without method calls (other than output).	
		S.C Write program code to satisfy method specifications using expressions, conditional statements, and iterative statements.	
	4.2 for Loops	3.C Write program code to satisfy method specifications using expressions, conditional statements, and iterative statements.	
CON-2		4.C Determine if two or more code segments yield equivalent results.	
		5.C Explain how the result of program code changes, given a change to the initial code.	
	4.3 Developing Algorithms Using Strings	2.C Determine the result or output based on the statement execution order in a code segment containing method calls.	
		3.C Write program code to satisfy method specifications using expressions, conditional statements, and iterative statements.	
	4.4 Nested Iteration	1.B Determine code that would be used to complete code segments.	
		SC Write program code to satisfy method specifications using expressions, conditional statements, and iterative statements.	
		5.C Explain how the result of program code changes, given a change to the initial code.	
	4.5 Informal Code Analysis	2.D Determine the number of times a code segment will execute.	
AP	Go to AP Classroom to assign the	Personal Progress Check for Unit 4.	

Review the results in class to identify and address any student misunderstandings.



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	4.2	Jigsaw As a whole class, look at a code segment containing iteration and the resulting output. Afterward, divide students into groups, and provide each group with a slightly modified code segment. After groups have determined how the result changes based on their modified segment, have them get together with students who investigated a different version of the code segment and share their conclusions.
2	4.1–4.4	Note-taking Provide students with a method that, when given an integer, returns the month name from a String that includes all the month names in order, each separated by a space. Have them annotate what each statement does in the method. Then, ask students to use their annotated method as a guide to write a similar method that, given a student number as input, returns the name of a student from a String containing the first name of all students in the class, each separated by a space.
3	4.5	Simplify the problem Provide students with several code segments containing iteration. For each segment, have students trace through the execution of a loop with smaller bounds to see what boundary cases are considered, and then use that information to determine the number of times each loop executes with the original bounds.

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 writing and analyzing program code.

() After completing this unit, students will have covered all of the necessary content for the Consumer Review Lab. The proposed class periods for this unit include time to complete the provided lab activities.

SUGGESTED SKILLS

1.B

Determine code that would be used to complete code segments.

2.B

Determine the result or output based on statement execution order in a code segment without method calls (other than output).

3.C

Write program code to satisfy method specifications using expressions, conditional statements, and iterative statements.



AVAILABLE RESOURCES

- Runestone Academy: AP CSA—Java Review: 7.2—While Loops
- Practice-It!: BJP4 Chapter 5: Program Logic and Indefinite Loops—Exercises 5.1–5.4
- The Exam > 2017 AP Computer Science A Exam Free-Response Question #3, Part B (Phrase)

TOPIC 4.1 while Loops

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.C

Represent iterative processes using a while loop.

ESSENTIAL KNOWLEDGE

CON-2.C.1

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

CON-2.C.2

In loops, the Boolean expression is evaluated before each iteration of the loop body, including the first. When the expression evaluates to true, the loop body is executed. This continues until the expression evaluates to false, whereupon the iteration ceases.

CON-2.C.3

A loop is an infinite loop when the Boolean expression always evaluates to true.

CON-2.C.4

If the Boolean expression evaluates to false initially, the loop body is not executed at all.

CON-2.C.5

Executing a return statement inside an iteration statement will halt the loop and exit the method or constructor.

continued on next page

LEARNING OBJECTIVE

CON-2.D

For algorithms in the context of a particular specification that does not require the use of traversals:

- Identify standard algorithms.
- Modify standard algorithms.
- Develop an algorithm.

ESSENTIAL KNOWLEDGE

CON-2.D.1

There are standard algorithms to:

- Identify if an integer is or is not evenly divisible by another integer
- Identify the individual digits in an integer
- Determine the frequency with which a specific criterion is met

CON-2.D.2

There are standard algorithms to:

- Determine a minimum or maximum value
- Compute a sum, average, or mode

SUGGESTED SKILLS

Write program code to satisfy method specifications using expressions, conditional statements, and iterative statements.

4.C

Determine if two or more code segments yield equivalent results.

5.C

Explain how the result of program code changes, given a change to the initial code.



AVAILABLE RESOURCES

- Runestone Academy: AP CSA—Java Review: 7.3—For Loops
- Practice-It!: BJP4 Chapter 2: Primitive Data and Definite Loops—Exercises 2.2, 2.3

TOPIC 4.2 for Loops

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.E

Represent iterative processes using a for loop.

ESSENTIAL KNOWLEDGE

CON-2.E.1

There are three parts in a for loop header: the initialization, the Boolean expression, and the increment. The increment statement can also be a decrement statement.

CON-2.E.2

In a for loop, the initialization statement is only executed once before the first Boolean expression evaluation. The variable being initialized is referred to as a loop control variable.

CON-2.E.3

In each iteration of a for loop, the increment statement is executed after the entire loop body is executed and before the Boolean expression is evaluated again.

CON-2.E.4

A for loop can be rewritten into an equivalent while loop and vice versa.

CON-2.E.5

"Off by one" errors occur when the iteration statement loops one time too many or one time too few.

TOPIC 4.3 Developing Algorithms Using Strings

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.F

For algorithms in the context of a particular specification that involves String objects:

- Identify standard algorithms.
- Modify standard algorithms.
- Develop an algorithm.

ESSENTIAL KNOWLEDGE

CON-2.F.1

There are standard algorithms that utilize String traversals to:

- Find if one or more substrings has a particular property
- Determine the number of substrings that meet specific criteria
- Create a new string with the characters reversed

SUGGESTED SKILLS

UNIT

Δ

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

3.C

Write program code to satisfy method specifications using expressions, conditional statements, and iterative statements.

-		ς.	
	_		
	-		
	-		
-		4	

AVAILABLE RESOURCES

- Practice-Itl: BJP4 Chapter 3: Parameters and Objects—Exercise 3.19
- Practice-It!: BJP4 Chapter 5: Program Logic and Indefinite Loops—Exercise 5.24
- CodingBat Java: String-2

SUGGESTED SKILLS

1.B

Determine code that would be used to complete code segments.

3.C

Write program code to satisfy method specifications using expressions, conditional statements, and iterative statements.

5.C

Explain how the result of program code changes, given a change to the initial code.



AVAILABLE LAB

 Classroom Resources > AP Computer Science A: Consumer Review Lab

AVAILABLE RESOURCES

- Runestone Academy: AP CSA—Java Review: 7.4—Nested For Loops
- Practice-It!: BJP4 Chapter 2: Primitive Data and Definite Loops—Exercises 2.4–2.15
- Past AP Free-Response Exam Questions on Methods and Control Structures on AP Question Bank

TOPIC 4.4 Nested Iteration

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.G

Represent nested iterative processes.

ESSENTIAL KNOWLEDGE

CON-2.G.1

Nested iteration statements are iteration statements that appear in the body of another iteration statement.

CON-2.G.2

When a loop is nested inside another loop, the inner loop must complete all its iterations before the outer loop can continue.

TOPIC 4.5 Informal Code Analysis

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.H

Compute statement execution counts and informal run-time comparison of iterative statements.

ESSENTIAL KNOWLEDGE

CON-2.H.1

A statement execution count indicates the number of times a statement is executed by the program.



Determine the number of times a code segment will execute.

UNIT

Δ

AP COMPUTER SCIENCE A

UNIT 5 Writing Classes



AP

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 5

Multiple-choice: ~25 questions Free-response: 2 questions

- Class
- Class: partial



5-7.5[%] AP EXAM WEIGHTING

~12-14 CLASS PERIODS

Writing Classes

BIG IDEA 1

Modularity MOD

 How can using a model of traffic patterns improve travel time?

BIG IDEA 2 Variables VAR

 How can programs be written to protect your bank account balance from being inadvertently changed?

BIG IDEA 3 Impact of Computing IOC

 What responsibility do programmers have for the consequences of programs they create, whether intentional or not?

<→ Developing Understanding

This unit will pull together information from all previous units to create new, user-defined reference data types in the form of classes. The ability to accurately model real-world entities in a computer program is a large part of what makes computer science so powerful. This unit focuses on identifying appropriate behaviors and attributes of real-world entities and organizing these into classes. Students will build on what they learn in this unit to represent relationships between classes through hierarchies, which appear in Unit 9.

The creation of computer programs can have extensive impacts on societies, economies, and cultures. The legal and ethical concerns that come with programs and the responsibilities of programmers are also addressed in this unit.

Building Computational Thinking Practices

Computer scientists use computers to study and model the world, and this requires designing program code to fit a given scenario. Spending time to plan out the program up front will shorten overall program development time. Practicing elements of the iterative development process early and often will help create more robust programs and avoid logical errors.

Because revisions to program code are rarely completed by the original author, documenting program code is very important. It allows programmers to more quickly understand how a code segment functions without having to analyze the program code itself. Well-written documentation includes a description of the behavior, as well as the initial conditions that must be met for a code segment to work as intended.

Sometimes students struggle to explain why a code segment will not compile or work as intended. It helps to have students work with a collaborative partner to practice verbally explaining the issues. The ability to explain why a code segment isn't working correctly assists in the development of solutions.

Preparing for the AP Exam

On the free-response section of the exam, students are required to design program code that demonstrates the functionality available for an object of a new class, based on the prompt's specification. This process includes identifying the attributes (data) that define an object of a class and the behaviors (methods) that define what an object of a class can do. Students should have many opportunities in this course to design their own classes based on program specifications and on observations of real-world objects.

Once the new class is designed, students should be able to implement program code to create a new type by creating a class. The behaviors of an object are expressed through writing *methods* in the class, which include expressions, conditional statements, and iterative statements. By being able to create their own classes, programmers are not limited to the existing classes provided within the Java libraries and can therefore represent their own ideas through classes.



UNIT AT A GLANCE

ring rstandinç			Class Periods
Endui Undei	Торіс	Suggested Skills	-12-14 CLASS PERIODS
0D-2 0D-3	5.1 Anatomy of a Class	1.A Determine an appropriate program design to solve a problem or accomplish a task (<i>not assessed</i>).	
22		1.B Determine code that would be used to complete code segments.	
	5.2 Constructors	1.C Determine code that would be used to interact with completed program code.	
MOD-2		3. Write program code to define a new type by creating a class.	
	5.3 Documentation with Comments	Describe the initial conditions that must be met for a program segment to work as intended or described.	
	5.4 Accessor Methods	3.B Write program code to define a new type by creating a class.	
		5.B Explain why a code segment will not compile or work as intended.	
	5.5 Mutator Methods	3.B Write program code to define a new type by creating a class.	
		4.B Identify errors in program code.	
	5.6 Writing Methods	1.B Determine code that would be used to complete code segments.	
		3.B Write program code to define a new type by creating a class.	
	5.7 Static Variables and Methods	3.B Write program code to define a new type by creating a class.	
		5. Describe the behavior of a given segment of program code.	

continued on next page

UNIT AT A GLANCE (cont'd)

uring erstanding			Class Periods
End	Торіс	Suggested Skills	~12-14 CLASS PERIODS
	5.8 Scope and Access	3.B Write program code to define a new type by creating a class.	
AR-1		5.B Explain why a code segment will not compile or work as intended.	
4	5.9 this Keyword	2C Determine the result or output based on the statement execution order in a code segment containing method calls.	
10C-1	5.10 Ethical and Social Implications of Computing Systems	Curricular requirement, not assessed on the AP Exam	
AP	Go to AP Classroom to assign the Review the results in class to identify the results in class to	ne Personal Progress Check for Unit 5. tify and address any student misunderstandings.	

UNIT

5



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	5.1	Kinesthetic learning Have students break into groups of 4–5 to play board games. Ask them to play the game for about 10 minutes. While they play the game, they should keep track of the various nouns they encounter and actions that happen as part of the game. The nouns can be represented in the computer as classes, and the actions are the behaviors. At the end of game play, ask students to create UML diagrams for the identified classes.
2	5.3	Marking the text Present students with specifications, and have them highlight or underline any preconditions (both implicit and explicit) that exist for the method to function. This includes information about parameters, such as object references not being null.
3	5.4–5.7	Create a plan When asked to write a method, have students write an outline using pseudocode with paper and pencil. Then, go through it step-by-step with sample input to ensure that the process is correct and to determine if any additional information is needed before beginning to program a solution on the computer.
4	5.7	Paraphrase Provide students with several example classes that utilize static variables for unique identification numbers or for counting the number of objects that have been created, but do not provide any description or documentation for the code. Have students spend time creating objects and calling the static methods to investigate how the static variables behave, then have them document the code appropriately to describe how each class utilizes static variables and methods.

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 writing and analyzing program code.

TOPIC 5.1 Anatomy of a Class

Required Course Content

ENDURING UNDERSTANDING

MOD-2

Programmers use code to represent a physical object or nonphysical concept, real or imagined, by defining a class based on the attributes and/or behaviors of the object or concept.

LEARNING OBJECTIVE

MOD-2.A

Designate access and visibility constraints to classes, data, constructors, and methods.

ESSENTIAL KNOWLEDGE

MOD-2.A.1

The keywords public and private affect the access of classes, data, constructors, and methods.

MOD-2.A.2

The keyword private restricts access to the declaring class, while the keyword public allows access from classes outside the declaring class.

MOD-2.A.3

Classes are designated public.

MOD-2.A.4

Access to attributes should be kept internal to the class. Therefore, instance variables are designated as private.

MOD-2.A.5

Constructors are designated public.

MOD-2.A.6

Access to behaviors can be internal or external to the class. Therefore, methods can be designated as either public or private.

SUGGESTED SKILL

Determine an appropriate program design to solve a problem or accomplish a task.

UNIT

5



Determine code that would be used to complete code segments.



AVAILABLE RESOURCES

- Runestone Academy: AP CSA—Java Review: 2.7—Parts of a Java Class
- Practice-It!: BJP4 Chapter 8: Classes— Self-Check 8.1–8.7
- Classroom Resources > Object-Oriented Design

a task.

AP Computer Science A Course and Exam Description

Writing Classes

ENDURING UNDERSTANDING

MOD-3

When multiple classes contain common attributes and behaviors, programmers create a new class containing the shared attributes and behaviors forming a hierarchy. Modifications made at the highest level of the hierarchy apply to the subclasses.

LEARNING OBJECTIVE

MOD-3.A

Designate private visibility of instance variables to encapsulate the attributes of an object.

ESSENTIAL KNOWLEDGE

MOD-3.A.1

Data encapsulation is a technique in which the implementation details of a class are kept hidden from the user.

MOD-3.A.2

When designing a class, programmers make decisions about what data to make accessible and modifiable from an external class. Data can be either accessible or modifiable, or it can be both or neither.

MOD-3.A.3

Instance variables are encapsulated by using the private access modifier.

MOD-3.A.4

The provided accessor and mutator methods in a class allow client code to use and modify data.

Writing Classes

TOPIC 5.2 Constructors

Required Course Content

ENDURING UNDERSTANDING

MOD-2

Programmers use code to represent a physical object or nonphysical concept, real or imagined, by defining a class based on the attributes and/or behaviors of the object or concept.

LEARNING OBJECTIVE

MOD-2.B

Define instance variables for the attributes to be initialized through the constructors of a class.

ESSENTIAL KNOWLEDGE

MOD-2.B.1

An object's state refers to its attributes and their values at a given time and is defined by instance variables belonging to the object. This creates a "has-a" relationship between the object and its instance variables.

MOD-2.B.2

Constructors are used to set the initial state of an object, which should include initial values for all instance variables.

MOD-2.B.3

Constructor parameters are local variables to the constructor and provide data to initialize instance variables.

MOD-2.B.4

When a mutable object is a constructor parameter, the instance variable should be initialized with a copy of the referenced object. In this way, the instance variable is not an alias of the original object, and methods are prevented from modifying the state of the original object.

MOD-2.B.5

When no constructor is written, Java provides a no-argument constructor, and the instance variables are set to default values.



Determine code that would be used to interact with completed program code.

UNIT

5

3.B

Write program code to define a new type by creating a class.

AVAILABLE RESOURCES

- Runestone Academy: AP CSA—Java Review: 2.7—Parts of a Java Class
- Practice-It!: BJP4 Chapter 8: Classes— Exercises 8.14–8.22



SUGGESTED SKILL

Describe the initial conditions that must be met for a program segment to work as intended or described.

UNIT

5



AVAILABLE RESOURCE

 External Resource > Zeroturnaround: Reasons, Tips and Tricks for Better Java Documentation

TOPIC 5.3 Documentation with Comments

Required Course Content

ENDURING UNDERSTANDING

MOD-2

Programmers use code to represent a physical object or nonphysical concept, real or imagined, by defining a class based on the attributes and/or behaviors of the object or concept.

LEARNING OBJECTIVE

MOD-2.C

Describe the functionality and use of program code through comments.

ESSENTIAL KNOWLEDGE

MOD-2.C.1

Comments are ignored by the compiler and are not executed when the program is run.

MOD-2.C.2

Three types of comments in Java include /* */, which generates a block of comments, //, which generates a comment on one line, and /** */, which are Javadoc comments and are used to create API documentation.

MOD-2.C.3

A precondition is a condition that must be true just prior to the execution of a section of program code in order for the method to behave as expected. There is no expectation that the method will check to ensure preconditions are satisfied.

MOD-2.C.4

A postcondition is a condition that must always be true after the execution of a section of program code. Postconditions describe the outcome of the execution in terms of what is being returned or the state of an object.

MOD-2.C.5

Programmers write method code to satisfy the postconditions when preconditions are met.

TOPIC 5.4 Accessor Methods

Required Course Content

ENDURING UNDERSTANDING

MOD-2

Programmers use code to represent a physical object or nonphysical concept, real or imagined, by defining a class based on the attributes and/or behaviors of the object or concept.

LEARNING OBJECTIVE

MOD-2.D

Define behaviors of an object through non-void methods without parameters written in a class.

ESSENTIAL KNOWLEDGE

MOD-2.D.1

An accessor method allows other objects to obtain the value of instance variables or static variables.

MOD-2.D.2

A non-void method returns a single value. Its header includes the return type in place of the keyword void.

MOD-2.D.3

In non-void methods, a return expression compatible with the return type is evaluated, and a copy of that value is returned. This is referred to as "return by value."

MOD-2.D.4

When the return expression is a reference to an object, a copy of that reference is returned, not a copy of the object.

MOD-2.D.5

The return keyword is used to return the flow of control to the point immediately following where the method or constructor was called.

continued on next page

SUGGESTED SKILLS

UNIT

5

Write program code to define a new type by creating a class.

5.B

Explain why a code segment will not compile or work as intended.

AVAILABLE LABS

- Classroom Resources >
 AP Computer Science A: Data Lab
 - AP Computer Science A: Celebrity Lab

AVAILABLE RESOURCE

 Practice-It!: BJP4 Chapter 8: Classes— Exercises 8.14–8.22

Writing Classes

LEARNING OBJECTIVE

MOD-2.D

Define behaviors of an object through non-void methods without parameters written in a class.

ESSENTIAL KNOWLEDGE

MOD-2.D.6

The toString method is an overridden method that is included in classes to provide a description of a specific object. It generally includes what values are stored in the instance data of the object.

MOD-2.D.7

If System.out.print or System.out. println is passed an object, that object's toString method is called, and the returned string is printed.

Writing Classes

TOPIC 5.5 Mutator Methods

Required Course Content

ENDURING UNDERSTANDING

MOD-2

Programmers use code to represent a physical object or nonphysical concept, real or imagined, by defining a class based on the attributes and/or behaviors of the object or concept.

LEARNING OBJECTIVE

MOD-2.E

Define behaviors of an object through void methods with or without parameters written in a class.

ESSENTIAL KNOWLEDGE

MOD-2.E.1

A void method does not return a value. Its header contains the keyword void before the method name.

MOD-2.E.2

A mutator (modifier) method is often a void method that changes the values of instance variables or static variables. SUGGESTED SKILLS

UNIT

5

Write program code to define a new type by creating a class.

4.B

Identify errors in program code.



AVAILABLE RESOURCE Practice-It!: BJP4 Chapter 8: Classes— Exercises 8.14–8.22



SUGGESTED SKILLS 1.B

Determine code that would be used to complete code segments.

UNIT

5



Write program code to define a new type by creating a class.



AVAILABLE RESOURCES

- Practice-It!: BJP4 Chapter 3: Parameters and Objects— Exercises 3.1–3.22
- Practice-It!: BJP4 Chapter 8: Class— Exercises 8.14–8.22

TOPIC 5.6 Writing Methods

Required Course Content

ENDURING UNDERSTANDING

MOD-2

Programmers use code to represent a physical object or nonphysical concept, real or imagined, by defining a class based on the attributes and/or behaviors of the object or concept.

LEARNING OBJECTIVE

MOD-2.F

Define behaviors of an object through non-void methods with parameters written in a class.

ESSENTIAL KNOWLEDGE

MOD-2.F.1

Methods can only access the private data and methods of a parameter that is a reference to an object when the parameter is the same type as the method's enclosing class.

MOD-2.F.2

Non-void methods with parameters receive values through parameters, use those values, and return a computed value of the specified type.

MOD-2.F.3

It is good programming practice to not modify mutable objects that are passed as parameters unless required in the specification.

MOD-2.F.4

When an actual parameter is a primitive value, the formal parameter is initialized with a copy of that value. Changes to the formal parameter have no effect on the corresponding actual parameter.

continued on next page

Writing Classes

LEARNING OBJECTIVE

MOD-2.F

Define behaviors of an object through non-void methods with parameters written in a class.

ESSENTIAL KNOWLEDGE

MOD-2.F.5

When an actual parameter is a reference to an object, the formal parameter is initialized with a copy of that reference, not a copy of the object. If the reference is to a mutable object, the method or constructor can use this reference to alter the state of the object.

MOD-2.F.6

Passing a reference parameter results in the formal parameter and the actual parameter being aliases. They both refer to the same object.



SUGGESTED SKILLS 3.B

UNIT

5

Write program code to define a new type by creating a class.

5.A

Describe the behavior of a given segment of program code.

TOPIC 5.7 Static Variables and Methods

Required Course Content

ENDURING UNDERSTANDING

MOD-2

Programmers use code to represent a physical object or nonphysical concept, real or imagined, by defining a class based on the attributes and/or behaviors of the object or concept.

LEARNING OBJECTIVE

MOD-2.G

Define behaviors of a class through static methods.

ESSENTIAL KNOWLEDGE

MOD-2.G.1

Static methods are associated with the class, not objects of the class.

MOD-2.G.2

Static methods include the keyword static in the header before the method name.

MOD-2.G.3

Static methods cannot access or change the values of instance variables.

MOD-2.G.4

Static methods can access or change the values of static variables.

MOD-2.G.5

Static methods do not have a this reference and are unable to use the class's instance variables or call non-static methods.

MOD-2.H

Define the static variables that belong to the class.

MOD-2.H.1

Static variables belong to the class, with all objects of a class sharing a single static variable.

continued on next page
Writing Classes

LEARNING OBJECTIVE

MOD-2.H Define the static variables that belong to the class.

ESSENTIAL KNOWLEDGE

MOD-2.H.2

Static variables can be designated as either public or private and are designated with the static keyword before the variable type.

MOD-2.H.3

Static variables are used with the class name and the dot operator, since they are associated with a class, not objects of a class.



SUGGESTED SKILLS 5.B

Explain why a code segment will not compile or work as intended.

UNIT

5

3.B

Write program code to define a new type by creating a class.

TOPIC 5.8 Scope and Access

Required Course Content

ENDURING UNDERSTANDING

VAR-1

To find specific solutions to generalizable problems, programmers include variables in their code so that the same algorithm runs using different input values.

LEARNING OBJECTIVE

VAR-1.G

Explain where variables can be used in the program code.

ESSENTIAL KNOWLEDGE

VAR-1.G.1

Local variables can be declared in the body of constructors and methods. These variables may only be used within the constructor or method and cannot be declared to be public or private.

VAR-1.G.2

When there is a local variable with the same name as an instance variable, the variable name will refer to the local variable instead of the instance variable.

VAR-1.G.3

Formal parameters and variables declared in a method or constructor can only be used within that method or constructor.

VAR-1.G.4

Through method decomposition, a programmer breaks down a large problem into smaller subproblems by creating methods to solve each individual subproblem.

Writing Classes

торіс 5.9 this Keyword

Required Course Content

ENDURING UNDERSTANDING

VAR-1

To find specific solutions to generalizable problems, programmers include variables in their code so that the same algorithm runs using different input values.

LEARNING OBJECTIVE

VAR-1.H

Evaluate object reference expressions that use the keyword this.

ESSENTIAL KNOWLEDGE

VAR-1.H.1

Within a non-static method or a constructor, the keyword this is a reference to the current object—the object whose method or constructor is being called.

VAR-1.H.2

The keyword this can be used to pass the current object as an actual parameter in a method call.

SUGGESTED SKILL

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

UNIT

5

AVAILABLE RESOURCE

≡

 Past AP Free-Response Exam Questions on Class on AP Question Bank

AVAILABLE RESOURCE

 Classroom Resources > Ethical Use of the Computer

UNIT

TOPIC 5.10 Ethical and Social Implications of Computing Systems

Required Course Content

ENDURING UNDERSTANDING

IOC-1

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

LEARNING OBJECTIVE

IOC-1.A

Explain the ethical and social implications of computing systems.

ESSENTIAL KNOWLEDGE

IOC-1.A.1

System reliability is limited. Programmers should make an effort to maximize system reliability.

IOC-1.A.2

Legal issues and intellectual property concerns arise when creating programs.

IOC-1.A.3

The creation of programs has impacts on society, economies, and culture. These impacts can be beneficial and/or harmful.

AP COMPUTER SCIENCE A





AP

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 6

Multiple-choice: ~15 questions Free-response: 2 questions

- Array and ArrayList (Array only)
- Array and ArrayList (Array only): partial





BIG IDEA 1 Variables VAR

 How can programs leverage volcano data to make predictions about the date of the next eruption?

BIG IDEA 2 Control CON

 How can knowing standard algorithms be useful when solving new problems?

Developing Understanding

This unit focuses on data structures, which are used to represent collections of related data using a single variable rather than multiple variables. Using a data structure along with iterative statements with appropriate bounds will allow for similar treatment to be applied more easily to all values in the collection. Just as there are useful standard algorithms when dealing with primitive data, there are standard algorithms to use with data structures. In this unit, we apply standard algorithms to arrays; however, these same algorithms are used with ArrayLists and 2D arrays as well. Additional standard algorithms, such as standard searching and sorting algorithms, will be covered in the next unit.

Building Computational Thinking Practices

3.D 4.B

Students should be able to implement program code to create, traverse, and manipulate elements in a 1D array. Traversing elements of a 1D array can be accomplished in multiple ways. Programmers need to make decisions about which loop structure is most effective given the problem they are trying to solve. Some loop structures, such as the enhanced for loop, only allow programmers to examine the data stored in a 1D array structure, while other loop structures allow the data to be manipulated.

Students should be able to identify and correct errors related to traversing and manipulating 1D array structures. A common run-time error that programmers experience is an out-of-bounds error, which occurs when the program tries to access an element that is beyond the range of elements in a collection. Students should double-check the values of the index being used on at least the initial and final loop iterations to ensure that they aren't out of bounds.

Preparing for the AP Exam

A specific iterative structure is commonly used to traverse an array: starting at the beginning and moving toward the last element in the array. When students are asked to determine the result of program code that contains arrays, they often use this same iterative structure. Knowing this iterative structure can be helpful when students use tracing to determine what an algorithm is doing. Students can follow this traversal for the first few iterations and apply that pattern to the remaining elements in the array.

When preparing for the free-response questions, students should be come familiar with how existing algorithms work, rather than just memorizing the program code. Have students write out an algorithm on paper and test it using manipulatives. This allows students to experience the algorithm on a deeper level than if they simply program it. A strong understanding of how existing algorithms work allows programmers to make modifications to those algorithms to accomplish similar tasks.

UNIT AT A GLANCE

UNIT

6

nding			
uring lersta			Class Periods
Cnd	Торіс	Suggested Skills	~6-8 CLASS PERIODS
VAR-2	6.1 Array Creation and Access	1.C Determine code that would be used to interact with completed program code.	
		Write program code to create, traverse, and manipulate elements in 1D array or ArrayList objects.	
	6.2 Traversing Arrays	2.B Determine the result or output based on statement execution order in a code segment without method calls (other than output).	
		Write program code to create, traverse, and manipulate elements in 1D array or ArrayList objects.	
		4.B Identify errors in program code.	
	6.3 Enhanced for Loop for Arrays	3.D Write program code to create, traverse, and manipulate elements in 1D array or ArrayList objects.	
		4.C Determine if two or more code segments yield equivalent results.	
CON-2	6.4 Developing Algorithms Using Arrays	1.B Determine code that would be used to complete code segments.	
		S.D. Write program code to create, traverse, and manipulate elements in 1D array or ArrayList objects.	
		Describe the initial conditions that must be met for a program segment to work as intended or described.	
AP	Go to AP Classroom to assign the Review the results in class to identif	Personal Progress Check for Unit 6. y and address any student misunderstandings.	



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	6.1	Diagramming Provide students with several prompts to create and access elements in an array. After they have determined the code for each prompt, have students draw a memory diagram that shows references and the arrays they point to. Have students update the diagram with each statement to demonstrate how changing the contents through one array reference effects all the array references for this array.
2	6.2	Error analysis Provide students with several error-ridden code segments containing array traversals along with the expected output of each segment. Ask them to identify any errors that they see on paper and to suggest fixes to provide the expected output. Have them type up their solutions in an IDE to verify their work.
3	6.3	Think-pair-share Ask students to consider two program code segments that are meant to yield the same result: one using a traditional for loop and one using a for each loop. Have them take a few minutes to think independently about whether the two segments accomplish the same result and, if not, what changes could be made in order for that to happen. Then, ask students to work with their partners to come up with situations where it would make sense to use one type of loop over the other before sharing with the whole class.
4	6.4	Pair programming Have students use pair programming to solve an array-based free-response question. Have one student be the driver for Part A while the other navigates, then have them switch for Part B. Once they are done, have partners switch solutions with another group and work through the scoring guidelines to "grade" that solution. Spend time as a class discussing the different approaches students used.

SUGGESTED SKILLS

Determine code that would be used to interact with completed program code.

UNIT

6

3.D

Write program code to create, traverse, and manipulate elements in 1D array or ArrayList objects.

AVAILABLE RESOURCES

- Runestone Academy: AP CSA—Java Review: 8.1—Arrays in Java
- Practice-It!: BJP4 Chapter 7: Array— Self-Check 7.1–7.9
- CodingBat Java: Array-1

TOPIC 6.1 Array Creation and Access

Required Course Content

ENDURING UNDERSTANDING

VAR-2

To manage large amounts of data or complex relationships in data, programmers write code that groups the data together into a single data structure without creating individual variables for each value.

LEARNING OBJECTIVE

VAR-2.A

Represent collections of related primitive or object reference data using onedimensional (1D) array objects.

ESSENTIAL KNOWLEDGE

VAR-2.A.1

The use of array objects allows multiple related items to be represented using a single variable.

VAR-2.A.2

The size of an array is established at the time of creation and cannot be changed.

VAR-2.A.3

Arrays can store either primitive data or object reference data.

VAR-2.A.4

When an array is created using the keyword new, all of its elements are initialized with a specific value based on the type of elements:

- Elements of type int are initialized to 0
- Elements of type double are initialized to 0.0
- Elements of type boolean are initialized to false
- Elements of a reference type are initialized to the reference value null. No objects are automatically created

VAR-2.A.5

Initializer lists can be used to create and initialize arrays.

continued on next page

LEARNING OBJECTIVE

VAR-2.A

Represent collections of related primitive or object reference data using onedimensional (1D) array objects.

ESSENTIAL KNOWLEDGE

VAR-2.A.6

Square brackets ([]) are used to access and modify an element in a 1D array using an index.

VAR-2.A.7

The valid index values for an array are 0 through one less than the number of elements in the array, inclusive. Using an index value outside of this range will result in an ArrayIndexOutOfBoundsException being thrown.

SUGGESTED SKILLS 2.B

Determine the result or output based on statement execution order in a code segment without method calls (other than output).

3.D

Write program code to create, traverse, and manipulate elements in 1D array or ArrayList objects.

4.B

Identify errors in program code.

AVAILABLE RESOURCES

- Runestone Academy: AP CSA—Java Review: 8.3—Using a For Loop to Loop through an Array
- CodingBat Java: Array-2
- Practice-It!: BJP4 Chapter 7: Arrays— Exercise 7.1–7.18

TOPIC 6.2 Traversing Arrays

Required Course Content

ENDURING UNDERSTANDING

VAR-2

To manage large amounts of data or complex relationships in data, programmers write code that groups the data together into a single data structure without creating individual variables for each value.

LEARNING OBJECTIVE

VAR-2.B

Traverse the elements in a 1D array.

ESSENTIAL KNOWLEDGE

VAR-2.B.1

Iteration statements can be used to access all the elements in an array. This is called traversing the array.

VAR-2.B.2

Traversing an array with an indexed for loop or while loop requires elements to be accessed using their indices.

VAR-2.B.3

Since the indices for an array start at 0 and end at the number of elements - 1, "off by one" errors are easy to make when traversing an array, resulting in an ArrayIndexOutOfBoundsException being thrown.

TOPIC 6.3 Enhanced for Loop for Arrays

Required Course Content

ENDURING UNDERSTANDING

VAR-2

To manage large amounts of data or complex relationships in data, programmers write code that groups the data together into a single data structure without creating individual variables for each value.

LEARNING OBJECTIVE

VAR-2.C

Traverse the elements in a 1D array object using an enhanced for loop.

ESSENTIAL KNOWLEDGE

VAR-2.C.1

An enhanced for loop header includes a variable, referred to as the enhanced for loop variable.

VAR-2.C.2

For each iteration of the enhanced for loop, the enhanced for loop variable is assigned a copy of an element without using its index.

VAR-2.C.3

Assigning a new value to the enhanced for loop variable does not change the value stored in the array.

VAR-2.C.4

Program code written using an enhanced for loop to traverse and access elements in an array can be rewritten using an indexed for loop or a while loop.

SUGGESTED SKILLS

Write program code to create, traverse, and manipulate elements in 1D array or ArrayList objects.

UNIT

6

4.C

Determine if two or more code segments yield equivalent results.

AVAILABLE RESOURCES

- Runestone Academy: AP CSA—Java Review:
 8.2—Looping with the For-Each Loop
- Practice-It!: BJP4 Chapter 7: Arrays— Exercises 7.1–7.18

SUGGESTED SKILLS 1.B

Determine code that would be used to complete code segments.

3.D

Write program code to create, traverse, and manipulate elements in 1D array or ArrayList objects.

5.D

Describe the initial conditions that must be met for a program segment to work as intended or described.



AVAILABLE RESOURCES

- Runestone Academy: AP CSA—Java Review: 8.13—Free-Response Questions
- CodingBat Java: Array 3
- Practice-It!: BJP4 Chapter 7: Array— Exercises 7.1–7.18

TOPIC 6.4 Developing Algorithms Using Arrays

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.I

For algorithms in the context of a particular specification that requires the use of array traversals:

- Identify standard algorithms.
- Modify standard algorithms.
- Develop an algorithm.

ESSENTIAL KNOWLEDGE

CON-2.I.1

There are standard algorithms that utilize array traversals to:

- Determine a minimum or maximum value
- Compute a sum, average, or mode
- Determine if at least one element has a particular property
- Determine if all elements have a particular property
- Access all consecutive pairs of elements
- Determine the presence or absence of duplicate elements
- Determine the number of elements meeting specific criteria

CON-2.I.2

There are standard array algorithms that utilize traversals to:

- Shift or rotate elements left or right
- Reverse the order of the elements

AP COMPUTER SCIENCE A

UNIT 7 ArrayList



AP

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 7

Multiple-choice: ~15 questions Free-response: 1 question

 Array and ArrayList (ArrayList focus)

UNIT

BIG IDEA 1 Variables VAR

Why is an ArrayList more appropriate for storing your music playlist, while an array might be more appropriate for storing your class schedule?

BIG IDEA 2 Control CON

 How can we use statement execution counts to choose appropriate algorithms?

BIG IDEA 3 Impact of Computing IOC

 What personal data is currently being collected, and how?

Developing Understanding

As students learned in Unit 6, data structures are helpful when storing multiple related data values. Arrays have a static size, which causes limitations related to the number of elements stored, and it can be challenging to reorder elements stored in arrays. The ArrayList object has a dynamic size, and the class contains methods for insertion and deletion of elements, making reordering and shifting items easier. Deciding which data structure to select becomes increasingly important as the size of the data set grows, such as when using a large real-world data set.

In this unit, students will also learn about privacy concerns related to storing large amounts of personal data and about what can happen if such information is compromised.

Building Computational Thinking Practices 2.C 2.D 3.D 5.C

Students need to consider the impact using ArrayList rather than an array has on the structure of their program code. This includes considering the use of ArrayList methods and the flexibility of a structure with a dynamic size. For instance, the use of an ArrayList will require students to analyze program code that uses method calls.

Providing students with practice writing programs for data sets of undetermined sized—or at least larger than they would be able to analyze easily by hand-presents a more relevant and realistic experience with data. Additionally, this requires students to focus more on the algorithm and ensuring that it will work in all situations rather than on an individual result.

With larger data sets, programmers become concerned with the amount of time it will take for their program code to run. Students should have practice determining the number of times a code segment executes; this can help them gain an idea of how long it will take to run a program on a data set of a given size.

Preparing for the AP Exam

When writing solutions to free-response questions that involve the use of an ArrayList, students are often asked to insert or delete elements from the ArrayList. In these cases, adjustments will need to be made to the loop counter to account for skipping an element or attempting to access elements that no longer exist.

Students may also be asked to traverse multiple data structures simultaneously. These data structures may be a mixture of array and ArrayList objects. As such, it is very easy for students to become confused about how elements are accessed and manipulated within each structure. Additionally, the size of the data structures may not be the same. In these situations, it is best to have different index variables and bounds checks for each structure to avoid accessing elements that are out-of-bounds.



UNIT AT A GLANCE

ring rstanding			Class Periods
Endui Unde	Торіс	Suggested Skills	~10-12 CLASS PERIODS
VAR-2	7.1 Introduction to ArrayList	1.B Determine code that would be used to complete code segments.	
		3.D Write program code to create, traverse, and manipulate elements in 1D array or ArrayList objects.	
	7.2 ArrayList Methods	2.C Determine the result or output based on the statement execution order in a code segment containing method calls.	
		Write program code to create, traverse, and manipulate elements in 1D array or ArrayList objects.	
	7.3 Traversing ArrayLists	2.C Determine the result or output based on the statement execution order in a code segment containing method calls.	
		Write program code to create, traverse, and manipulate elements in 1D array or ArrayList objects.	
	7.4 Developing Algorithms Using ArrayLists	3.D Write program code to create, traverse, and manipulate elements in 1D array or ArrayList objects.	
-		4.A Use test-cases to find errors or validate results.	
CON-2	7.5 Searching	3.D Write program code to create, traverse, and manipulate elements in 1D array or ArrayList objects.	
		5.C Explain how the result of program code changes, given a change to the initial code.	
	7.6 Sorting	2.D Determine the number of times a code segment will execute.	
100-1	7.7 Ethical Issues Around Data Collection	Curricular requirement, not assessed on the AP Exam	
AP	Go to AP Classroom to assign the	Personal Progress Check for Unit 7.	

Review the results in class to identify and address any student misunderstandings.



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	7.2	Predict and compare Have students look at the code they wrote to solve the free-response question in Unit 6 (or other code from Unit 6) on paper, and have them rewrite it using an ArrayList. Have them highlight the parts that need to be changed and determine how to change them. Then, have students type up the changes in an IDE and confirm that the program still works as expected.
2	7.1–7.5	Identify a subtask Have students read through an ArrayList-based free-response question in groups, and have them identify all subtasks. These subtasks could be conditional statements, iteration, or even other methods. Once the subtasks have been identified, divide the subtasks among the group members, and have students implement their given subtask. When all students are finished, have them combine the subtasks into a single solution.
3	7.5	Discussion group Discuss the algorithm necessary to search for the smallest value in an ArrayList. Without explaining what you are doing, change the Boolean expression so that it will find the largest value, and ask students to describe what the resulting algorithm will do. Then, change the algorithm to store and return the location of the largest value, and discuss the change.

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 writing and analyzing program code that uses real-world data sets.

SUGGESTED SKILLS 1.B

Determine code that would be used to complete code segments.

3.D

Write program code to create, traverse, and manipulate elements in 1D array or ArrayList objects.



AVAILABLE RESOURCES

- Java Quick Reference (see Appendix)
- Runestone Academy: AP CSA—Java Review: 9.7—The ArrayList Class
- Practice-It!: BJP4 Chapter 10: ArrayLists—Self-Check 10.2

TOPIC 7.1 Introduction to ArrayList

Required Course Content

ENDURING UNDERSTANDING

VAR-2

To manage large amounts of data or complex relationships in data, programmers write code that groups the data together into a single data structure without creating individual variables for each value.

LEARNING OBJECTIVE

VAR-2.D

Represent collections of related object reference data using ArrayList objects.

ESSENTIAL KNOWLEDGE

VAR-2.D.1

An ArrayList object is mutable and contains object references.

VAR-2.D.2

The ArrayList constructor ArrayList() constructs an empty list.

VAR-2.D.3

Java allows the generic type ArrayList<E>, where the generic type E specifies the type of the elements.

VAR-2.D.4

When ArrayList<E> is specified, the types of the reference parameters and return type when using the methods are type E.

VAR-2.D.5

ArrayList<E> is preferred over ArrayList because it allows the compiler to find errors that would otherwise be found at run-time.

TOPIC 7.2 ArrayList Methods

Required Course Content

ENDURING UNDERSTANDING

VAR-2

To manage large amounts of data or complex relationships in data, programmers write code that groups the data together into a single data structure without creating individual variables for each value.

LEARNING OBJECTIVE

VAR-2.D

Represent collections of related object reference data using ArrayList objects.

ESSENTIAL KNOWLEDGE

VAR-2.D.6

The ArrayList class is part of the java. util package. An import statement can be used to make this class available for use in the program.

VAR-2.D.7

The following ArrayList methods including what they do and when they are used—are part of the Java Quick Reference:

- int size() Returns the number of elements in the list
- boolean add(E obj) Appends obj to end of list; returns true
- void add(int index, E obj) Inserts obj at position index (0 <= index <= size), moving elements at position index and higher to the right (adds 1 to their indices) and adds 1 to size
- E get(int index) Returns the element at position index in the list
- E set(int index, E obj) Replaces the element at position index with obj; returns the element formerly at position index

SUGGESTED SKILLS

UNIT

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

3.D

Write program code to create, traverse, and manipulate elements in 1D array or ArrayList objects.



AVAILABLE RESOURCES

- Java Quick Reference (see Appendix)
- Practice-It!: BJP4 Chapter 10: ArrayLists—Exercises 10.2–10.17
- The Exam > 2017 AP Computer Science A Exam Free-Response Question #1, Part A (Digits)

continued on next page

LEARNING OBJECTIVE

VAR-2.D

Represent collections of related object reference data using ArrayList objects.

ESSENTIAL KNOWLEDGE

• E remove(int index)—Removes element from position index, moving elements at position index + 1 and higher to the left (subtracts 1 from their indices) and subtracts 1 from size; returns the element formerly at position index

TOPIC 7.3 Traversing ArrayLists

Required Course Content

ENDURING UNDERSTANDING

VAR-2

To manage large amounts of data or complex relationships in data, programmers write code that groups the data together into a single data structure without creating individual variables for each value.

LEARNING OBJECTIVE

VAR-2.E

For ArrayList objects:

- a. Traverse using a for or while loop
- b. Traverse using an enhanced for loop

ESSENTIAL KNOWLEDGE

VAR-2.E.1

Iteration statements can be used to access all the elements in an ArrayList. This is called traversing the ArrayList.

VAR-2.E.2

Deleting elements during a traversal of an ArrayList requires using special techniques to avoid skipping elements.

VAR-2.E.3

Since the indices for an ArrayList start at 0 and end at the number of elements - 1, accessing an index value outside of this range will result in an ArrayIndexOutOfBoundsException being thrown.

VAR-2.E.4

Changing the size of an ArrayList while traversing it using an enhanced for loop can result in a ConcurrentModificationException being thrown. Therefore, when using an enhanced for loop to traverse an ArrayList, you should not add or remove elements.

SUGGESTED SKILLS

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

UNIT

3.D

Write program code to create, traverse, and manipulate elements in 1D array or ArrayList objects.



AVAILABLE RESOURCES

- Runestone Academy: AP CSA—Java Review: 9.14—Looping through a List
- Practice-It!: BJP4 Chapter 10: ArrayLists—Exercises 10.2–10.17
- The Exam > 2018 AP Computer Science A Exam Free-Response Question #2 (WordPair)

SUGGESTED SKILLS 3.D

Write program code to create, traverse, and manipulate elements in 1D array or ArrayList objects.

4.A

Use test-cases to find errors or validate results.



AVAILABLE LAB

 Classroom Resources > AP Computer Science A: Data Lab

AVAILABLE RESOURCES

- Runestone Academy: AP CSA—Java Review: 9.13—Removing an Object at an Index
- Practice-It!: BJP4 Chapter 10: ArrayLists—Exercises 10.2–10.17
- The Exam >
 - 2017 AP Computer Science A Exam Free-Response Question 31, Part B (Digits)
 - Past AP Free-Response Exam Questions on Array/ ArrayList on AP Question Bank

TOPIC 7.4 Developing Algorithms Using ArrayLists

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.J

For algorithms in the context of a particular specification that requires the use of ArrayList traversals:

- Identify standard algorithms.
- Modify standard algorithms.
- Develop an algorithm.

ESSENTIAL KNOWLEDGE

CON-2.J.1

There are standard ArrayList algorithms that utilize traversals to:

- Insert elements
- Delete elements
- Apply the same standard algorithms that are used with 1D arrays

CON-2.J.2

Some algorithms require multiple String, array, or ArrayList objects to be traversed simultaneously.

TOPIC 7.5 Searching

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.K

Apply sequential/linear search algorithms to search for specific information in array or ArrayList objects.

ESSENTIAL KNOWLEDGE

CON-2.K.1

There are standard algorithms for searching.

CON-2.K.2

Sequential/linear search algorithms check each element in order until the desired value is found or all elements in the array or ArrayList have been checked.

SUGGESTED SKILLS

Write program code to create, traverse, and manipulate elements in 1D array or ArrayList objects.

UNIT

5.C

Explain how the result of program code changes, given a change to the initial code.

AVAILABLE RESOURCES

- Practice-It!: BJP4 Chapter 10: ArrayLists—Exercises 10.2–10.17
- Runestone Academy: AP CSA—Java Review: 13—Searching and Sorting
- Practice-It!: BJP4 Chapter 13: Searching and Sorting

ArrayList

SUGGESTED SKILL 2.D

Determine the number of times a code segment will execute.



AVAILABLE LAB

 Classroom Resources > AP Computer Science A: Data Lab

AVAILABLE RESOURCES

- Runestone Academy: AP CSA—Java Review: 13—Searching and Sorting
- Practice-It!: BJP4 Chapter 13: Searching and Sorting—Self-Check 13.29 and 13.30
- Visualgo.net: Sorting
- Sorting.at

TOPIC 7.6 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.L

Apply selection sort and insertion sort algorithms to sort the elements of array or ArrayList objects.

CON-2.M

Compute statement execution counts and informal run-time comparison of sorting algorithms.

ESSENTIAL KNOWLEDGE

CON-2.L.1

Selection sort and insertion sort are iterative sorting algorithms that can be used to sort elements in an array or ArrayList.

CON-2.M.1

Informal run-time comparisons of program code segments can be made using statement execution counts.

TOPIC 7.7 Ethical Issues Around Data Collection

Required Course Content

ENDURING UNDERSTANDING

IOC-1

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

LEARNING OBJECTIVE

IOC-1.B

Explain the risks to privacy from collecting and storing personal data on computer systems.

ESSENTIAL KNOWLEDGE

IOC-1.B.1

When using the computer, personal privacy is at risk. Programmers should attempt to safeguard personal privacy.

IOC-1.B.2

Computer use and the creation of programs have an impact on personal security. These impacts can be beneficial and/or harmful. AVAILABLE RESOURCES

UNIT

- Classroom Resources >
 Ethical Use of the Computer
 - Ethical Issues: Internet Content Providers and Internet Service Providers

AP COMPUTER SCIENCE A

UNIT 8 2D Array



AP

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 8

Multiple-choice: ~10 questions Free-response: 1 question

2D Array



~10-12 CLASS PERIODS

2D Array

BIG IDEA 1

- Variables VARWhy might you want to
- use a 2D array to store the spaces on a game board or the pixels in a picture, rather than a 1D array or ArrayList?

BIG IDEA 2 Control CON

 Why does the order in which elements are accessed in 2D array traversal matter in some situations?

Developing Understanding

In Unit 6, students learned how 1D arrays store large amounts of related data. These same concepts will be implemented with two-dimensional (2D) arrays in this unit. A 2D array is most suitable to represent a table. Each table element is accessed using the variable name and row and column indices. Unlike 1D arrays, 2D arrays require nested iterative statements to traverse and access all elements. The easiest way to accomplished this is in row-major order, but it is important to cover additional traversal patterns, such as back and forth or column-major.

Building Computational Thinking Practices

Students should be able to determine the result of program code that traverses and manipulates the elements in a 2D array. Traversals of 2D arrays typically require a set of nested loops. Often the extra dimension of a 2D array is difficult for students to envision. Providing students with practice analyzing and tracing traversals of a 2D array, as well as providing them partial program code to complete, helps students take this more abstract concept and make it concrete and replicable.

Because 2D arrays are traversed using nested loops, the number of times a code segment executes is multiplied. In a nested loop, the inner loop must complete all iterations before the outer loop can continue. It helps to provide students with sample code that will print the values in a 2D array. Teachers can an IDE that shows access to 2D arrays visually and keeps track of the execution count.

Preparing for the AP Exam

The free-response portion of the exam always includes one question that requires students to write program code involving 2D arrays. Because 2D arrays are arrays where each element is an array, it is not uncommon for the question to require students to write solutions involving array or ArrayList objects as well.

While there is a specific nested structure to traverse elements in a 2D array in rowmajor order, this structure can be modified to traverse 2D arrays in other ways, such as column-major, by switching the nested iterative statements. Additional modifications can be made to traverse rows or columns in different ways, such as back and forth or up and down. However, when making these adjustments, students often neglect to adjust the bounds of the iterative statements appropriately. Students should practice traversing 2D arrays in these nonstandard ways, being sure to test the boundary conditions of the iterative statements, to be prepared for this type of freeresponse question.



UNIT AT A GLANCE

luring derstanding			Class Periods
CE	Topic	Suggested Skills	~10-12 CLASS PERIODS
	8.1 2D Arrays	1.B Determine code that would be used to complete code segments.	
VAR-2		1.C Determine code that would be used to interact with completed program code.	
		3.E Write program code to create, traverse, and manipulate elements in 2D array objects.	
VAR-2, CON-2	8.2 Traversing 2D Arrays	2.B Determine the result or output based on statement execution order in a code segment without method calls (other than output).	
		2.D Determine the number of times a code segment will execute.	
		3.E Write program code to create, traverse, and manipulate elements in 2D array objects.	
		4.A Use test-cases to find errors or validate results.	
AP	Go to AP Classroom to assign the Personal Progress Check for Unit 8. Review the results in class to identify and address any student misunderstandings.		



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	8.1	Using manipulatives Use different-sized egg cartons or ice cube trays with random compartments filled with small toys or candy. Create laminated cards with the code for the construction of, and access to, a 2D array, leaving blanks for the name and size dimensions. Have students fill in the missing code that would be used to represent the physical 2D array objects and access the randomly stored elements.
2	8.2	Activating prior knowledge When first introducing 2D arrays and row-major traversal, ask students which part of the nested for loop structure loops over a 1D array. Based on what they know about the traversal of 1D array structures, ask them to calculate the number of times the inner loop executes.
3	8.2	Sharing and responding As a class, create a set of test cases to be used with answers to a free-response question. Have students write their answers to the free-response question individually on paper. After exchanging solutions with another student, ask students to find errors or validate results of their peers' code by tracing the code with the developed test cases. Allow students an opportunity to provide feedback on the program code as well as the results of each test case.

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 writing and analyzing program code that represents 2D data sets.

() After completing this unit, students will have covered all of the necessary content for the Steganography Lab. The proposed class periods for this unit include time to complete the provided lab activities.

SUGGESTED SKILLS

1.B

Determine code that would be used to complete code segments.

1.C

Determine code that would be used to interact with completed program code.

3.E

Write program code to create, traverse, and manipulate elements in 2D array objects.

AVAILABLE LABS

- Classroom Resources >
- AP Computer Science A: Picture Lab
- AP Computer Science A: Steganography Lab

AVAILABLE RESOURCES

- Runestone Academy: AP CSA—Java Review: 10—Two-dimensional Arrays
- Practice-Itl: BJP4 Chapter 7: Arrays— Self-Check 7.31–7.35
- Classroom Resources > GridWorld Resources: A Curriculum Module for AP Computer Science

TOPIC 8.1 2D Arrays

Required Course Content

ENDURING UNDERSTANDING

VAR-2

To manage large amounts of data or complex relationships in data, programmers write code that groups the data together into a single data structure without creating individual variables for each value.

LEARNING OBJECTIVE

VAR-2.F

Represent collections of related primitive or object reference data using two-dimensional (2D) array objects.

ESSENTIAL KNOWLEDGE

VAR-2.F.1

2D arrays are stored as arrays of arrays. Therefore, the way 2D arrays are created and indexed is similar to 1D array objects.

EXCLUSION STATEMENT—(EK VAR-2.F.1): 2D array objects that are not rectangular are outside the scope of the course and AP Exam.

VAR-2.F.2

For the purposes of the exam, when accessing the element at arr[first][second], the first index is used for rows, the second index is used for columns.

VAR-2.F.3

The initializer list used to create and initialize a 2D array consists of initializer lists that represent 1D arrays.

VAR-2.F.4

The square brackets [row][col] are used to access and modify an element in a 2D array.

VAR-2.F.5

"Row-major order" refers to an ordering of 2D array elements where traversal occurs across each row, while "column-major order" traversal occurs down each column.

TOPIC 8.2 Traversing 2D Arrays

Required Course Content

ENDURING UNDERSTANDING

VAR-2

To manage large amounts of data or complex relationships in data, programmers write code that groups the data together into a single data structure without creating individual variables for each value.

LEARNING OBJECTIVE

VAR-2.G

For 2D array objects:

- Traverse using nested for loops.
- Traverse using nested enhanced for loops.

ESSENTIAL KNOWLEDGE

VAR-2.G.1

Nested iteration statements are used to traverse and access all elements in a 2D array. Since 2D arrays are stored as arrays of arrays, the way 2D arrays are traversed using for loops and enhanced for loops is similar to 1D array objects.

VAR-2.G.2

Nested iteration statements can be written to traverse the 2D array in "row-major order" or "column-major order."

VAR-2.G.3

The outer loop of a nested enhanced for loop used to traverse a 2D array traverses the rows. Therefore, the enhanced for loop variable must be the type of each row, which is a 1D array. The inner loop traverses a single row. Therefore, the inner enhanced for loop variable must be the same type as the elements stored in the 1D array.

SUGGESTED SKILLS 2.B

Determine the result or output based on statement execution order in a code segment without method calls (other than output).

UNIT

B

2.D

Determine the number of times a code segment will execute.

3.E

Write program code to create, traverse, and manipulate elements in 2D array objects.

4.A

Use test-cases to find errors or validate results.

AVAILABLE LABS

- Classroom Resources >
- AP Computer
 Science A: Picture
 Lab
- AP Computer
 Science A:
 Steganography Lab

AVAILABLE RESOURCES

- Runestone Academy: AP CSA—Java Review: 10.7—Looping through a 2D Array
- Practice-It!: BJP4 Chapter 7: Arrays— Exercises 7.19–7.19
- The Exam >
 - 2017 AP Computer Science A Exam Free-Response Question #4 (Position)
 - 2018 AP Computer Science A Exam Free-Response Question #4 (ArrayTester)
 - Past AP Exam Questions on 2D Arrays on AP Question Bank

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.N

For algorithms in the context of a particular specification that requires the use of 2D array traversals:

- Identify standard algorithms.
- Modify standard algorithms.
- Develop an algorithm.

ESSENTIAL KNOWLEDGE

CON-2.N.1

When applying sequential/linear search algorithms to 2D arrays, each row must be accessed then sequential/linear search applied to each row of a 2D array.

CON-2.N.2

All standard 1D array algorithms can be applied to 2D array objects.
AP COMPUTER SCIENCE A

UNIT 9 Inheritance



AP

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 9

Multiple-choice: ~15 questions Free-response: 2 questions

- Class
- Class: partial



5–10[%] AP EXAM WEIGHTING

Inheritance

BIG IDEA 1 Modularity MOD

- How might the use of inheritance help in writing a program that simulates crops being grown in a virtual world?
- How does inheritance make programs more versatile?

↔ Developing Understanding

Creating objects, calling methods on the objects created, and being able to define a new data type by creating a class are essential understandings before moving into this unit. One of the strongest advantages of Java is the ability to categorize classes into hierarchies through *inheritance*. Certain existing classes can be extended to include new behaviors and attributes without altering existing code. These newly created classes are called *subclasses*. In this unit, students will learn how to recognize common attributes and behaviors that can be used in a *superclass* and will then create a hierarchy by writing subclasses to extend a superclass. Recognizing and utilizing existing hierarchies will help students create more readable and maintainable programs.

Building Computational Thinking Practices

Students can design hierarchies by listing the attributes and behaviors for each object and pulling common elements into a superclass, leaving unique attributes and behaviors in the subclass. By creating a hierarchy system, students only need to write common program code one time, reducing potential errors and implementation time. This also allows for changes to be made more easily, because they can be made at the superclass level in the hierarchy and automatically apply to subclasses.

During the development of a program, programmers often use comments to describe the behavior of a given segment of program code and to describe the initial conditions that are used. Students who develop the skill of explaining why program code does not work, such as methods being overloaded improperly or superclass objects attempting to call subclass methods, are much better equipped to foresee and avoid these hierarchy issues.

Preparing for the AP Exam

One question on the free-response section of the exam will require students to write a class. This class could be part of an inheritance relationship. When overriding superclass methods in a subclass, method signatures must be the same. This includes the number, type, and order of any parameters of the overridden method. It is important for students to recognize when a method should be overridden, as well as when they can and should use methods from the superclass. Students who duplicate code unnecessarily will not earn full points on this question.

Students will be asked to analyze program code that uses inheritance. In many cases, students struggle with determining whether a method is available to be called by an object of a class. When a method is called on a subclass object, the method that is executed is determined during run-time. If the subclass does not contain the called method, the superclass method will automatically be executed.



UNIT AT A GLANCE

iring erstanding			Class Periods
Cude	Торіс	Suggested Skills	~14-15 CLASS PERIODS
MOD-3	9.1 Creating Superclasses and Subclasses	 1.A Determine an appropriate program design to solve a problem or accomplish a task (<i>not assessed</i>). 3.B Write program code to define a new type by 	
		creating a class.	
	9.2 Writing Constructors for Subclasses	3.B Write program code to define a new type by creating a class.	
		5.A Describe the behavior of a given segment of program code.	
	9.3 Overriding Methods	3.B Write program code to define a new type by creating a class.	
		5.D Describe the initial conditions that must be met for a program segment to work as intended or described.	
	9.4 super Keyword	1.C Determine code that would be used to interact with completed program code.	
		3.B Write program code to define a new type by creating a class.	
	9.5 Creating References Using Inheritance Hierarchies	3.A Write program code to create objects of a class and call methods.	
		5.B Explain why a code segment will not compile or work as intended.	
	9.6 Polymorphism	3.A Write program code to create objects of a class and call methods.	
		5.B Explain why a code segment will not compile or work as intended.	
	9.7 Object Superclass	1.C Determine code that would be used to interact with completed program code.	
		3.B Write program code to define a new type by creating a class.	

Go to **AP Classroom** to assign the **Personal Progress Check** for Unit 9. Review the results in class to identify and address any student misunderstandings.

AP

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	9.1	Activating prior knowledge Have students review what they know about classes, methods, and the scope of variables by having them write a class based on specifications that can easily be extended by subclasses. This class will become the superclass for subclasses they write later in the unit.
2	9.2–9.4	Create a plan Given a class design problem that requires the use of multiple classes in an inheritance hierarchy, students identify the common attributes and behaviors among these classes and write these into a superclass. Any additional information that does not belong in the superclass will be categorized to determine the additional classes that might be necessary and what methods will need to be added or overridden in the subclasses.
3	9.4	Think aloud Provide students with a code segment that contains method calls using the super keyword. Have students describe the code segment out loud to themselves. Give students several individual statements that attempt to interact with the given code segment, and have them talk through each one, describing which statements would work and which ones would not, as well as the reasons why those statements wouldn't work.
4	9.5–9.6	Student response system Provide students with several statements where objects are created and the reference type and object type are different but related. Then provide students with calls to methods on these created objects. Use a student response system to have students determine whether each statement is legal, would result in a compile-time error, or would result in a run-time error.

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 writing and analyzing program code.

{} After completing this unit, students will have covered all of the necessary content for the Celebrity Lab. The proposed class periods for this unit include time to complete the provided lab activities.



SUGGESTED SKILLS

Determine an appropriate program design to solve a problem or accomplish a task.

3.B

Write program code to define a new type by creating a class.



AVAILABLE RESOURCES

- Runestone Academy: AP CSA—Java Review: 11.3—Inheritance
- Classroom Resources >
 Object-Oriented Design
 - An Introduction to Polymorphism in Java

TOPIC 9.1 Creating Superclasses and Subclasses

Required Course Content

ENDURING UNDERSTANDING

MOD-3

When multiple classes contain common attributes and behaviors, programmers create a new class containing the shared attributes and behaviors forming a hierarchy. Modifications made at the highest level of the hierarchy apply to the subclasses.

LEARNING OBJECTIVE

MOD-3.B

Create an inheritance relationship from a subclass to the superclass.

ESSENTIAL KNOWLEDGE

MOD-3.B.1

A class hierarchy can be developed by putting common attributes and behaviors of related classes into a single class called a superclass.

MOD-3.B.2

Classes that extend a superclass, called subclasses, can draw upon the existing attributes and behaviors of the superclass without repeating these in the code.

MOD-3.B.3

Extending a subclass from a superclass creates an "is-a" relationship from the subclass to the superclass.

MOD-3.B.4

The keyword extends is used to establish an inheritance relationship between a subclass and a superclass. A class can extend only one superclass.

TOPIC 9.2 Writing Constructors for Subclasses

Required Course Content

ENDURING UNDERSTANDING

MOD-3

When multiple classes contain common attributes and behaviors, programmers create a new class containing the shared attributes and behaviors forming a hierarchy. Modifications made at the highest level of the hierarchy apply to the subclasses.

LEARNING OBJECTIVE

MOD-3.B

Create an inheritance relationship from a subclass to the superclass.

ESSENTIAL KNOWLEDGE

MOD-3.B.5

Constructors are not inherited.

MOD-3.B.6

The superclass constructor can be called from the first line of a subclass constructor by using the keyword super and passing appropriate parameters.

MOD-3.B.7

The actual parameters passed in the call to the superclass constructor provide values that the constructor can use to initialize the object's instance variables.

MOD-3.B.8

When a subclass's constructor does not explicitly call a superclass's constructor using super, Java inserts a call to the superclass's no-argument constructor.

MOD-3.B.9

Regardless of whether the superclass constructor is called implicitly or explicitly, the process of calling superclass constructors continues until the Object constructor is called. At this point, all of the constructors within the hierarchy execute beginning with the Object constructor. SUGGESTED SKILLS

3.B

Write program code to define a new type by creating a class.

UNIT

9

5.A

Describe the behavior of a given segment of program code.

- Practice-It!: BJP4 Chapter 9: Inheritance and Interfaces—Self-Check 9.3
- Classroom Resources >
 An Introduction to Polymorphism in Java
 - Gradebook Project

SUGGESTED SKILLS 3.B

Write program code to define a new type by creating a class.

5.D

Describe the initial conditions that must be met for a program segment to work as intended or described.

AVAILABLE RESOURCES

- Runestone Academy: AP CSA—Java Review: 11.8—Overriding vs Overloading
- Practice-It!: BJP4
 Chapter 9: Inheritance and Interfaces—
 Exercises 9.4, 9.9
- Classroom Resources >
- An Introduction to Polymorphism in Java
- Gradebook Project

TOPIC 9.3 Overriding Methods

Required Course Content

ENDURING UNDERSTANDING

MOD-3

When multiple classes contain common attributes and behaviors, programmers create a new class containing the shared attributes and behaviors forming a hierarchy. Modifications made at the highest level of the hierarchy apply to the subclasses.

LEARNING OBJECTIVE

MOD-3.B

Create an inheritance relationship from a subclass to the superclass.

ESSENTIAL KNOWLEDGE

MOD-3.B.10

Method overriding occurs when a public method in a subclass has the same method signature as a public method in the superclass.

MOD-3.B.11

Any method that is called must be defined within its own class or its superclass.

MOD-3.B.12

A subclass is usually designed to have modified (overridden) or additional methods or instance variables.

MOD-3.B.13

A subclass will inherit all public methods from the superclass; these methods remain public in the subclass.

TOPIC 9.4 Super Keyword

Required Course Content

ENDURING UNDERSTANDING

MOD-3

When multiple classes contain common attributes and behaviors, programmers create a new class containing the shared attributes and behaviors forming a hierarchy. Modifications made at the highest level of the hierarchy apply to the subclasses.

LEARNING OBJECTIVE

MOD-3.B

Create an inheritance relationship from a subclass to the superclass.

ESSENTIAL KNOWLEDGE

MOD-3.B.14

The keyword **super** can be used to call a superclass's constructors and methods.

MOD-3.B.15

The superclass method can be called in a subclass by using the keyword super with the method name and passing appropriate parameters.

SUGGESTED SKILLS

Determine code that would be used to interact with completed program code.

UNIT

9

3.B

Write program code to define a new type by creating a class.



- Runestone Academy: AP CSA—Java Review: 11.9—Using Super to Call an Overridden Method
- Classroom Resources >
 Gradebook Project
 - Inheritance and Polymorphism with Sudoku

SUGGESTED SKILLS 3.A

Write program code to create objects of a class and call methods.

5.B

Explain why a code segment will not compile or work as intended.



AVAILABLE RESOURCES

- Practice-It!: BJP4
 Chapter 9: Inheritance and Interfaces—Self-Check 9.8, 9.10
- Classroom Resources > Gradebook Project

TOPIC 9.5 Creating References Using Inheritance Hierarchies

Required Course Content

ENDURING UNDERSTANDING

MOD-3

When multiple classes contain common attributes and behaviors, programmers create a new class containing the shared attributes and behaviors forming a hierarchy. Modifications made at the highest level of the hierarchy apply to the subclasses.

LEARNING OBJECTIVE

MOD-3.C

Define reference variables of a superclass to be assigned to an object of a subclass in the same hierarchy.

ESSENTIAL KNOWLEDGE

MOD-3.C.1

When a class **S** "is-a" class **T**, **T** is referred to as a superclass, and **S** is referred to as a subclass.

MOD-3.C.2

If S is a subclass of T, then assigning an object of type S to a reference of type T facilitates polymorphism.

MOD-3.C.3

If S is a subclass of T, then a reference of type T can be used to refer to an object of type T or S.

MOD-3.C.4

Declaring references of type T, when S is a subclass of T, is useful in the following declarations:

- Formal method parameters
- arrays T[] var ArrayList<T> var

TOPIC 9.6 Polymorphism

Required Course Content

ENDURING UNDERSTANDING

MOD-3

When multiple classes contain common attributes and behaviors, programmers create a new class containing the shared attributes and behaviors forming a hierarchy. Modifications made at the highest level of the hierarchy apply to the subclasses.

LEARNING OBJECTIVE

MOD-3.D

Call methods in an inheritance relationship.

ESSENTIAL KNOWLEDGE

MOD-3.D.1

Utilize the Object class through inheritance.

MOD-3.D.2

At compile time, methods in or inherited by the declared type determine the correctness of a non-static method call.

MOD-3.D.3

At run-time, the method in the actual object type is executed for a non-static method call.

SUGGESTED SKILLS 3.A

Write program code to create objects of a class and call methods.

UNIT

9

5.B

Explain why a code segment will not compile or work as intended.



AVAILABLE LAB

 Classroom Resources > AP Computer Science A: Celebrity Lab

- Runestone Academy: AP CSA—Java Review: 11.15—Polymorphism
- Practice-It!: BJP4 Chapter 9: Inheritance and Interfaces—Self-Check 9.9



SUGGESTED SKILLS

Determine code that would be used to interact with completed program code.



Write program code to define a new type by creating a class.



AVAILABLE LAB

 Classroom Resources > AP Computer Science A: Celebrity Lab

AVAILABLE RESOURCE

 Java Quick Reference (see Appendix)

TOPIC 9.7 Object Superclass

Required Course Content

ENDURING UNDERSTANDING

MOD-3

When multiple classes contain common attributes and behaviors, programmers create a new class containing the shared attributes and behaviors forming a hierarchy. Modifications made at the highest level of the hierarchy apply to the subclasses.

LEARNING OBJECTIVE

MOD-3.E

Call Object class methods through inheritance.

ESSENTIAL KNOWLEDGE

MOD-3.E.1

The Object class is the superclass of all other classes in Java.

MOD-3.E.2

The Object class is part of the java.lang package

MOD-3.E.3

The following Object class methods and constructors—including what they do and when they are used—are part of the Java Quick Reference:

- boolean equals(Object other)
- String toString()

MOD-3.E.4

Subclasses of Object often override the equals and toString methods with class-specific implementations.

AP COMPUTER SCIENCE A

UNIT 10 Recursion



AP

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)



5-7.5[%] AP EXAM WEIGHTING



Recursion

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?

↔ Developing Understanding

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

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.



Recursion

UNIT AT A GLANCE

uring Ierstanding			Class Periods
End	Topic	Suggested Skills	~3-5 CLASS PERIODS
CON-2	10.1 Recursion	1.B Determine code that would be used to complete code segments.	
		5.A Describe the behavior of a given segment of program code.	
	10.2 Recursive Searching and Sorting	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.	
AP	Go to AP Classroom to assign the Review the results in class to identif	Personal Progress Check for Unit 10. y and address any student misunderstandings.	



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	Sharing and responding 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.
2	10.1	Look for a pattern 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.
3	10.2	Code tracing 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.

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.



Recursion

SUGGESTED SKILLS

Determine code that would be used to complete code segments.



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.

Recursion

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.

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

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

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

UNIT

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.

- 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