

AP COMPUTER SCIENCE A

UNIT 1

Primitive Types



2.5–5%
AP EXAM WEIGHTING



~8–10
CLASS PERIODS

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

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

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

Personal Progress Check 1

Multiple-choice: ~25 questions

Primitive Types



Developing Understanding

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?

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

2.A 4.B 5.A 5.B

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

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

SAMPLE INSTRUCTIONAL ACTIVITIES

The sample activities on this page are optional and are offered to provide possible ways to incorporate instructional approaches into the classroom. They were developed in partnership with teachers from the AP community to share ways that they approach teaching some of the topics in this unit. Please refer to the Instructional Approaches section beginning on p. 159 for more examples of activities and strategies.

Activity	Topic	Sample Activity
1	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

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.



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

VAR-1.A

Create string literals.

ESSENTIAL KNOWLEDGE

VAR-1.A.1

A string literal is enclosed in double quotes.

TOPIC 1.2

Variables and Data Types

SUGGESTED SKILLS

1.A

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

1.B

Determine code that would be used to complete code segments.



AVAILABLE RESOURCE

- [Runestone Academy: AP CSA—Java Review: 3—Variables](#)

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

1.B

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

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.



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., $\text{arr}[x++]$) 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

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.



AVAILABLE RESOURCE

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

