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

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	**ESSENTIAL KNOWLEDGE**
VAR-1.B Identify the most appropriate data type category for a particular specification.	**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 Declare variables of the correct types to represent primitive data.	**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.

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.

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.

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.