Java Fundamentals
4-3: Data Types and Operators
Practice Activities

Lesson Objectives:

- Use primitive data types in Java code
- Specify literals for the primitive types and for Strings
- Demonstrate how to initialize variables
- Describe the scope rules of a method
- Recognize when an expression requires a type conversion
- Apply casting in Java code
- Use arithmetic operators
- Use the assignment operator
- Use a method from the Math class
- Access a Math class method from the Java API

Vocabulary:

Identify the vocabulary word for each definition below.

Variables	Named primitive or object storage mechanisms defined in a program. The assigned value may or may not (constants) change.
Arithmetic Operators	Symbols are used to do addition, subtraction, multiplication, division, and modular arithmetic in math expressions and formulas.
Primitive Data Types	The group of Java data types that do not use the keyword new when declared or initialized. Primitive Data Types store the value in the same place in memory as the variable name.
Byte	The smallest java primitive type (1 byte) that can hold an integer value.
long	This data type (8 bytes) is the largest integer type.
coding conventions	The formatting and naming standards that most programmers follow.
int	This Java primitive data type (4 bytes) can hold integer values.
double	This Java primitive data type (8 bytes) is the largest primitive that can hold a decimal value.
Initialization	When a variable is assigned a value for the first time.
float	This Java primitive data type (4 bytes) can be initialized with a decimal number preceding letter f. Example: float x = 3.5f;
literal	Can be any number, text, or other information that represents a value; used to initialize a primitive type.

Declaration	A Java statement when a variable is defined but not necessarily assigned a value. Example: int x;
operator precendence	This word describes the mathematical precedence that a variable has in a Java program.
char	A java primitive data type (2 bytes) that can hold single character values. Example: “a”, “#”, or “X”
	Used to describe the block of code where a variable exists in a program. A block of code is denoted by {}.
	The process of explicitly modifying one data type to become a different data type.
	A concept where a number is always rounded down to the nearest integer.
	The equals sign “=” used in a Java statement to assign a value to a variable.
	The process of modifying one data type to become a different data type, this may be implicit or explicit.
	A Java primitive data type (2 bytes) that holds integer numbers within a shorter range than an int.
	A one-bit java primitive type that can hold the value true or false.

Try It/Solve It:

1. Write a program that will take in the base and height of a triangle and calculate and display the area of the triangle using the formula below.

   $$A = \frac{1}{2}bh$$

2. Write the following math formulas in Java. You will need to use methods from the Math class as well as nesting of methods and parentheses to force the order of operations to correctly calculate the answer. Assume that all the variables in the formulas have already been declared and initialized.

   a. $a = \frac{\sqrt{x^5 - 6}}{4}$ a = Math.sqrt(Math.pow(x, 3) - 6) / 4;

   b. $b = x^y - 6^x$ b = Math.pow(x, y) - Math.pow(6, x);

   c. $c = 4cos(\frac{z}{5}) - sinx^2$ c = 4 * Math.cos(z / 5.0) - Math.sin(Math.pow(x, 2));

   d. $d = x^4 - \sqrt{6x - y^3}$ d = Math.pow(x, 4) - Math.sqrt(6 * x - Math.pow(y, 3));

   e. $e = \frac{1}{y - \frac{1}{x - 2^y}}$ e = 1 / (y - (1 / (x - 2 * y)));

   f. $f = 7(cos(\sqrt{5 - sin\sqrt{3x - 4}}))$ f = 7 * Math.cos(Math.sqrt(5 - Math.sqrt(3 * x - 4)));

3. A bus holds 45 people. The school will only use a bus if they can fill it completely. The rest of the people will ride in vans. Write a program that will take in the number of people that are signed up to go on a field trip. Have the program print the number of busses necessary and then total number of people that will need to ride in vans.

4. Write true or false on the blanks in the program below to show the value of the boolean variable true_false as the program executes.

```
int i=5;
int j=6;
boolean true_false;
true_false=(j<5);                                   false
true_false=(j>3);                                   true
true_false=(j<i);                                   false
true_false=(i<5);                                   false
true_false=(j<=5);                                  false
true_false=(6<6);                                   false
true_false=(i!=j);                                  true
true_false=(i==j || i<50);                          true
true_false=(i==j && i<50);                          false
true_false=(i>j || true_false && j>=4);             n/a
true_false=(!(i<2 && j==5));                        _____
true_false=!true_false;                             _____
```

5. Explain why each of the declarations in the second list are wrong.

```
boolean gameOver = false;
int students=50,classes=3;
double sales_tax;
short number1;
```

```
int 2beOrNot2be;
float price index;
double lastYear'sPrice;
long class;
```

6. Explain why each of the declarations in the second list do not follow conventions for variable names.

```
int cadence=3, speed=55,
gear=4;
final double SALES_TAX=.06;
double gearRatio=.5;
int currentGear=5;
```

```
int c=3,s=55,g=4;
final double salesTax=.06;
double gearratio=.05,Gear=4;
int current_gear;
```