

## Java Fundamentals

### 2-13: Java Variables and Data Types

### Practice Activities

#### Lesson Objectives:

- Describe variables
- Describe Java simple types
- Define arithmetic operators
- Describe relational and logical operators
- Describe assignment operators

#### Vocabulary:

Identify the vocabulary word for each definition below.

	A lexical unit used to express a relation, such as equality or greater than, between two expressions.
	A set of data with values having predefined characteristics.
	Boolean operators (AND, OR and NOT).
	A place in memory where data of a specific type can be stored for later retrieval and use.
	A lexical unit used to perform basic mathematical operations by taking two operands and returning the result of the mathematical calculation.

#### Try It/Solve It:

1. In Alice 3, you will create an animation of a child exercising. You will also declare variables.
  - a. Create a world with a child. Save the project as "Child Exercising".
  - b. Have the child wave. Assign the value of the wave to "I'm happy".
  - c. Have the child say "I would like to exercise today."
  - d. Have the child do several exercises (side stretches, touch toes, jumping jacks). After exercising, have the child stand and say, "I'm all done exercising."
  - e. Change the code so that before the child exercises, you declare a variable of type Integer called numSets. Set the default value to 3. Save your animation.
  - f. Use this value to control how many sets of exercises the child does.
  - g. Have the child wave and say goodbye at the end of the animation.

- In Alice 3, create an animation with an alien riding a vehicle of your choosing. Save the project as "Alien Landing". Program the alien to fly through the air, and then spin as it slowly lowers to the ground. Declare a local variable to count the number of times the alien should spin and lower to the ground.
- In Alice 3, create an animation entitled "Flying Birds". Add two birds to the scene, each in a random location, and one tree in the center of the scene. Using variables, an if-else statement, and relational operators, test the distance of each bird to the tree by programming the birds to act in accordance with the following textual storyboard (Review slides on Relational Operators for programming hints):

```

    If bird 1 is currently a shorter distance to the tree than bird 2, then,
        bird 1 flies to the tree
    else
        bird 2 flies to the tree

```

- Complete the following Java Syntax Review Sheet:

Construct	Syntax
Assignment operators	
Arithmetic operators	
Equality operators	
Relational operators	
Logical operators	

- What are the results of the following code?

```

class basicOperators2 {
    //using arithmetic operators and variables
    public static void main(String[] args) {
        int a = 1+ 3;
        int b = a * 3;
        int c = b / 4;
        int d = c - a;
        int e = -d;
        System.out.println("a = " + a);
        System.out.println("b = " + b);
        System.out.println("c = " + c);
        System.out.println("d = " + d);
        System.out.println("e = " + e);
    }
}

```

- What are the results of the following code?

```

class Test {
    public static void main(String[] args) {
        int a = 10;
        int b = 20;
        System.out.println("a == b = " + (a == b) );
        System.out.println("a != b = " + (a != b) );
        System.out.println("a > b = " + (a > b) );
        System.out.println("a < b = " + (a < b) );
        System.out.println("b >= a = " + (b >= a) );
        System.out.println("b <= a = " + (b <= a) );
    }
}

```

8. This example demonstrates the NOT operator. Review the code, then fill in the blanks below with either “true” or “false”.

```
class BoolNotDemo {  
    public static void main(String[] args){  
        int x = 2;  
        int y = 1;  
        boolean bl;  
        bl = !(x > y); // bl is false  
        System.out.println("x is not greater than y:"+bl);  
        bl = !(y > x); // bl is true  
        System.out.println("y is not greater than x:"+bl);  
    }  
}
```

Fill in the blanks:

x is not greater than y :

y is not greater than x :

9. This example demonstrates assignment syntax. Review the code, then fill in the blanks below with the results.

```
class AssignmentDemo2{  
    public static void main(String[] args) {  
        int x=5;  
        int y=10;  
        x += y;  
        System.out.println("The += result is:"+ x);  
        x -= y;  
        System.out.println("The -= result is:"+ x);  
        x *= y;  
        System.out.println("The *= result is:"+ x);  
        x /= y;  
        System.out.println("The /= result is"+ x);  
    }  
}
```

Fill in the blanks:

The += result is:

The -= result is:

The \*= result is:

The /= result is: