

# Introduction to Java

## Notes Packet #1

**Objectives:** By the completion of this packet, students should be able to

- Understand and use variables of the int and double data types.
- Understand and use assignment operators.
- Understand and use numeric literals.
- Understand and use math operators.
- Understand the effect of math operations on ints and mixed-data type expressions.
- Understand how to cast a double to an int.

### Variables and Assignment Statements.

A variable is \_\_\_\_\_

Our rules for naming variables \_\_\_\_\_

\_\_\_\_\_

Warning. Java is case-sensitive.

Two types of variables are ints and doubles. \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Let's look at an example of code that uses a variable.

```
public class First {  
    public static void main( String [] args ) {  
        int x; _____  
        x = 8; _____  
        System.out.println( "The variable x has a value of " + x );  
    }  
}
```

An assignment statement \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

The assignment operator is \_\_\_\_\_.



A variable can be declared and initialized in one statement.

---

Multiple variables (of the same type) can be declared in one statement.

---

**Numeric Literals** are the actual numbers used throughout the code. Numeric literals also have data types. For example:

```
double z = 5 * 2.1 - 3.0;
```

---

---

**Mathematical Operations.** Mathematical operations are evaluated using the same rules of precedence you learned in middle school: \_\_\_\_\_

---

For example, what is value of x? \_\_\_\_\_

```
double x;
```

```
x = 4.0 + 3.0 / 2.0 - 1.0;
```

**Caution When Dividing ints.** The result of any operation involving two ints (whether variables or literals) is an int. When dividing two ints, \_\_\_\_\_

---

<pre>int a; a = -29 / 10;</pre> <p>What is the value of <i>a</i>?</p>	<pre>int b; b = 3 / 4;</pre> <p>What is the value of <i>b</i>?</p>	<pre>double c; c = 387 / 100;</pre> <p>What is the value of <i>c</i>?</p>
---	--	---



**Working with ints and doubles.** We are going to examine some code snippets and determine if there are any errors in the code. If there is, then we identify the source of the problem. If there are no errors, then we specify the value of the variable(s).

double a = 5;	
int b = -3.6;	
double c = 1 / 2;	
double d = 1.0 / 2.0;	
double e = 1.0 / 2;	
int fred = 8.0;	
int g = 8, h = 5; double juliet = 10; double bo = (g + h) / juliet;	

**Casting** is the process of explicitly converting one data type to another (assuming the conversion makes sense). The cast operator (int) truncates the value toward zero. The casting operator has higher precedence than multiplication but lower than parentheses.

int x; x = (int) 7.8;	
int x; x = (int) (-6 + 0.2);	
int x; x = (int) 2.7 + 0.6;	

**Increment and Decrement Operators.** You can increase the value of an int or double by using the increment operator (++). To decrease its value, use --. For example:

```
int x = 5;
x++;
int y = 9;
y--;
System.out.println( x + ", " + y );
```

## Compound Assignment Operators

Java sometimes uses “short cuts” for certain common statements

Basic Version	Alternative Version
int x = 7; x = x + 4;	int x = 7; x += 4;
int y = 14; y = y - 2;	int y = 14; y -= 2;