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 **named storage location in memory.**

Our rules for naming variables **1) Start with a letter. 2) May follow with letters, numbers, and underscore. 3) No spaces. 4) No keywords (e.g. public, class, int, double).**

Warning. Java is case-sensitive.

Two types of variables are ints and doubles. **An int can store an integer between +/- 2 billion (approximately). A double can store a decimal within a wider range of values.**

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

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

An assignment statement **evaluates the expression to the right of the equals sign and copies/assigns that value to the variable on the left side of the equals sign.**

The assignment operator is **the equals sign =**.

A variable can be declared and initialized in one statement.

double y = 3.7;

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

int a, b, c;

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;
```

The above statement contains three numeric literals: 5 (an int), 2.1 (a double), and 3.0 (a double).

Mathematical Operations. Mathematical operations are evaluated using the same rules of precedence you learned in middle school: **parentheses, multiplication, division, addition, and subtraction.**

For example, what is value of x? **4.5**

```
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, **the result must be an int. Round the result toward zero.**

int a;
a = -29 / 10;
What is the value of *a*?
-2
	int b;
	b = 3 / 4;
	What is the value of *b*?
	0
		double c;
		c = 387 / 100;
		What is the value of *c*?
		3.0

You must evaluate the expression 387/100 first. Then assign the 3 to c.

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;	**5**
int b = -3.6;	**Error. Cannot assign a double to an int.**
double c = 1 / 2;	**0.0 The int 1 divided by the int 2 equals zero.**
double d = 1.0 / 2.0;	**0.5**
double e = 1.0 / 2;	**0.5**
int fred = 8.0;	**Error. 8.0 is a double and cannot assign it to an int.**

int g = 8, h = 5;
double juliet = 10;
double bo = (g + h)/ juliet;
	bo is 1.3 because g+h = 13 (an int) but juliet is a double so the expression is 13/10.0 which is 1.3

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;
	7

int x;
x = (int) (-6 + 0.2);
	-5

int x;
x = (int) 2.7 + 0.6;
	Error because only 2.7 is cast. Then 2 + 0.6 is a double which cannot be assigned to an int.

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 );
```

6, 8

Compound Assignment Operators

Java sometimes uses “short cuts” for certain common statements

Basic Version	Alternative Version

int x = 7;
x = x + 4;   **x is 11**
	int x = 7;
	x += 4; **x is 11**

int y = 14;
y = y – 2;   **y is 12**
	int y = 14;
	y -= 2; **y is 12**