Java Foundations September

- Java is a compiled language. We write the programs in a *.java file, and it compiles into a *.class file. The *.class is interpreted by a computer's Java Virtual Machine (JVM) into machine code for the particular computer.

- Java programs are called "classes". The first line of every program is:
```
public class ClassName{
```

- The curly braces, { }, are used in Java to group code together. The space between a matching pair of braces is called a scope.

- "Methods" are groups of statements that are executed together. Methods are called by using their name and parentheses.

- The main method is a special method. It is the method the computer looks for to execute a program. All Java programs have a main method somewhere.
```
public static void main(String[] args){
```

- Computers use a binary system to keep track of data, using 2 digits (0 and 1). For example
- Primative data types: int, double, boolean, char, and sometimes String.
    - int – these are integers. They hold numbers on the interval $[-2^{31},\ 2^{31}-1]$ They are 32 bit values.

    - double – your go-to for decimal (floating-point) values. About 16 decimal digits of accuracy. There can be some strange rounding errors in the last digit. These are 64 bit chunks of data.

    - boolean – true or false values. Used for flags or toggles, and in control statements (if statements). Likely 1 bit, but the size is not documented.

    - char – a single character, such as a letter, number or symbol. Indicated by single quotes ' ' (or apostrophes, if you please.). 16 bit Unicode.

    - String – a group of characters, grouped by double quotes " "

- Declaring variables. If you want a variable name to be active, you need to tell the computer what type of data it will store while creating the variable. For example:
```
int n;      //declares an integer variable named n.
n = 5;      //sets n to 5
int x = 7;  //declares an integer x, and assigns it the value 7
```

You only need to declare the variable once (tag it as an int or a double or whatever). After that, the variable name is valid and you can reference it. ***Declaring it again will cause an error.***

- Mathematics.
  - Java will follow order of operations. The basic operators are + - * /.
  - Watch out for integer division. An int divided by an int will return an int as the answer, by chopping off (truncating the decimals). Thus:
    5/2 → 2 7/5 → 1 100/30 → 3 9/10 → 0

    That last one is important! An unexpected 0 quotient can really mess up a program.
  - If you want a double answer, use a double somewhere in the division:
    5.0/2 → 2.5 7/5.0 → 1.4 100.0/30.0 → 3.33… 9/10. → 0.9
  - Other math functions can be found in the Math class. Google Java Math for more details, but these include Math.sqrt(n), Math.abs(n), Math.pow(a,b).

- Input. To read input from the console, use a Scanner object. Note that Scanner is imported from java.util.Scanner

```
Scanner in = new Scanner(System.in);
int inputNum = in.nextInt();
double inputDub = in.nextDouble();
String inputLine = in.nextLine();
```

- Classes and Objects

  A class is a model for some entity. There are 3 major components:

  Data (Instance Fields) – Every model will need to keep track of some information. This really defines what makes up your entity.

  Methods – These are the actions your entity will be able to perform. The verbs in your model.

  Constructor – This enables you to build instances (objects) of the class. Each object will have unique values for the instance fields, and the constructor is where those values come in and are assigned to the instance fields.

  An object is a usable, specific instance of a class. An analogy is that the classes are the blueprints for a car, and objects are the actual cars. Each real car may vary a bit (color, power seats, etc) but they are all based on the same blueprint.

- Check out the Class Syntax Primer document for details and good syntax examples.

- Methods can have a return type. This allows method calls to have a value. For example, the call Math.sqrt(5) returns a double, and we can save it to a double variable or treat it as a number. Methods we create can have the same behavior.

```
public double getArea(){
        return Math.PI * radius * radius;
}
```

  Note that return types can be any data type or any object type.
  If your method has a return type that is not void, it MUST return a value of that type.