**Objects and Classes**

**Classes**

Classes are basically descriptions of objects. They contain methods and variables that any object of that class would know. Classes in java are generally created in a new file, where the name of that file is *YourClassName.*java (YourClassName would be replaced with the name of your class). Here is an example class that we can use throughout the document:

public class Fruit

{

private String name;

public String getName()

{

return name;

}

public void setName(String n)

{

name = n;

}

}

This class is named Fruit. It has a variable to store its name. It has a method that will output the name and a method that can be used to set the name.

If you want to have a class that can run the program, you need to have the main method. This class below is called a driver program, which just means that it will run everything else:

public class Driver

{

public static void main (String[] args)

{

//some code

}

}

Note: // means comment in code. This means that the computer won’t pay attention to it. You can write anything you want in comments.

**Objects**

Objects are an instance of a class. Basically they are things that are defined by all of the specifications laid out in the class. One good metaphor is to think of the class as a blueprint, and the objects as the actual building created from the blueprint.

When you create a new object, it’s called instantiation. In general, there is one way to instantiate an object:

Object name = new Object();

“Object” is the type of object. Name is the name of the object, and it can be anything you want as long as it’s lowercase and starts with a letter (not a number or other character). In the next example, we can create an object of type Fruit, named apple:

Fruit apple = new Fruit();

We can then go on to set the name of apple to apple. You call every method using dot notation.

name.method();

apple.setName(“apple”);

Using programming language, apple isa Fruit. That means that apple knows everything that Fruit knows.

**Primitive Types**

There are some object types that are already defined in Java (and most programming languages). They can store one value and have a name.

int: Stores an integer value (whole number between –infinity to infinity)

double: Stores a decimal value between –infinity to infinity

boolean: Either true or false

String: A sequence of characters

Primitive types are instantiated a little differently than normal objects.

int x = 0;

double d = 0.0;

String s = “0”

boolean b = false;

**Projects**

Here are some projects that will make sure that you know everything you need to from this lesson:

Don’t forget, you can always ask and answer questions on the website, and if that doesn’t help, bring your questions to our next meeting.