**Methods**

**Why use methods?**

Methods are used to break code down into sections that can be called over and over again. It’s useful when there is something that you might have to use more than once, so you don’t have to write it multiple times. They also help organize the code.

Without using methods, you would write code like this:

int x = 0;

x++;

x = x\*2;

x = x-5;

int y = 7;

y++;

y = y\*2;

y = y-5;

int z = 2;

z++;

z = z\*2;

z = z-5;

System.out.println(x+y+z);

Instead, you could create a new method:

public int doStuff(int a)

{

Don’t worry about what everything means yet. The main point of this method is to show how they could be used. You’ll see what each part of the method means further down the page.

a++;

a = a\*2;

a = a-5;

return a

}

You can then use the method instead – It’s much simpler:

int x = 0;

int y = 7;

int z = 2;

System.out.println(doStuff(x), doStuff(y), doStuff(z));

**Method Structure**

Most methods follow a general structure. However, there are several different components that can change:

public void methodName()

{

//code

}

This is an example of a simple method. The most important things to focus on right now are the fact that there is a name for the method that you can decide, followed by open and closed parenthesis. Inside of two curly braces is the code that would be executed every time that the method is called.

public int anotherMethod(int x)

{

return x%2;

}

This method is slightly different. It has “int x” inside of the parenthesis. This means that when you call the method somewhere else, you can put your own variable there and that will be edited in the method. For example, using the method above:

int y = 5;

int newVar = anotherMethod(y);

System.out.println(newVar) 🡪 //prints out 1

This is also different because it has int instead of void. Putting something other than void just means that the method outputs something. You can store the output in a variable like we did above. We’ll go over this a little more below.

**Method Keywords**

public: This means that the method can be used by any other class

static: This means that the method is a class method – the whole class does the method and not individual objects of the class. Methods that don’t have the static keyword are instance methods – each object does this method.

void: The method doesn’t return anything

private: The method can only be accessed within the class

return: This is used within the method and tells the method that it is outputting whatever comes next

For practice with creating methods and method keywords, go to the website.

**Common Methods**

There are several methods you will come across frequently when you are coding:

* System.out.println() - This method is used to print something to the console. It’s a good way to check if your program is working step by step
* toString() – This shows the string representation of any object. It is very helpful if you want to print an object to see what the code is doing. Without this, you will get a weird combination of letters and numbers that make no sense.
* equals() – this shows if two objects have the same value
* Integer.parseInt() – turns a string version of an integer into an actual integer.
* Double.parseDouble – turns a string version of a double into an actual double.
* Math.random() – outputs a random decimal value between zero and one. You can multiply and add to this value to get it between a certain range.

**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.