The purpose of this lab is to get your through the first steps on using Java

1. Know what the Java API is.
2. Get used to Eclipse – which is the editor will be using to write and run java classes
3. Create and some simple java classes – in a package.

**Part 1 – Java API**

**The Java API is a library of java classes that do common things that you don’t want to rewrite from scratch. The API for java is available on the Oracle website. This is where all the existing java classes are listed. The Java API is also sometimes referred to as Java API Specification. Let’s take a quick look at it – before digging into creating our own java classes.**The Java API is at:

<http://docs.oracle.com/javase/8/docs/api/>

The classes are grouped into packages. An example of a package is java.applet.

1. On the left hand side top, see the list of **packages** in the API.
2. On the left hand side underneath package, see the list of **all classes** in the api. If you know you want to use a class, but don’t know what package it’s in, you just use this list.
3. Pick a class to look at as an example:
   1. Click on the Package “javax.swing” on top left window
   2. On the lower left window, scroll down from “interfaces” until you see “Classes”. Find JButton class and open it in the big main frame.

There is a lot of info stored about the class – most of which won’t mean much for now.

* + 1. How many “Constructors” does JButton have?

1. Try another packge – find the java.util package
   1. Click on the Package “javax.util” on top left window
   2. Find the “Arrays” class
   3. Look at the class details in the right hand window. Page down to methods and find at least 3 methods that the Arrays class has.

**PART 2 – Creating your first java class**

**Eclipse**

Eclipse is just an editing environment for helping you to create and run java code. If you are completely new to Eclipse, look at the Eclipse video in Webcourses if you want extra help on how to use the Eclipse screen.

Open up Eclipse on your machine.

You may be prompted to set the **workspace** (this is where you code will be stored). If so, just set to **your** u drive (or wherever you want to store your code)

**Creating a java class**

Create a java project: In Eclipse – do File/New – java project. Give it a name.

Create a package within the project: Right click on “src” folder and create a new package. Call it com.lab1.test

Create your first class: From the package, create a new class. Call it HelloWorld.

This class won’t have any attributes.. it’s just a simple class to run things.

First put a comment header block into your new java class at the top to explain what it is e.g.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* HelloWorld

\* Author:

\* Date:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Write java code in your Helloword class to print “helloworld” out to the console –

* You will need a “Main” method – as discussed in class in your code as an entry point to your code. (this method can also be auto-generated for you in Eclipse as an option when you are creating the class).

Save your code into the src/package directory of your project.

Run your java class : by selecting Run/run from the menu or clicking on green run button arrow.

Check the console in the bottom pane to see the “print” output.

**PART 3 – Create a Class - Vehicle**

Create a new class (Using File/ New/ Class.. ) in the same directory as the others.

Call this class “Vehicle” to represent a vehicle

In your “Vehicle” class, first add your comment block at the top. Then, add attributes that represent obvious things you might want to store about it.. including as Owner name, registration number, maximum speed, colour, whether it is an automatic or not, number of wheels.

Add a **constructor** to your Vehicle class to set up new Vehicle objects, and that sets up the “owner name” attribute with a value.

Change the main method in your HelloWord class so that it creates a new Vehicle object (i.e. *instantiates* an object).

You can’t “see” anything - as creating objects isn’t actually outputting anything. Add a print statement to your Vehicle constructor to output the owner name allocated to that object. Now create more Vehicle objects.

Add **a second** constructor to your Vehicle class – that sets up all attributes with initial values. Add a System.out.println to this constructor to print out those values.

Now instantiate more Vehicle objects, this time using the 2nd constructor you created.

**PART 4 – toString()**

There is a special method called toString() in java which we’ll look at again. For now, add a method called “toString” to your Vehicle class. This method will return a String result.

Put in java code into the method that creates a String called “content”. Set content to be a the values of the attributes of the class.. put formatted with nice text e.g. “This vehicle has owner name X and is of colour Y etc”

In your main method , instantiate a new Vehicle object and add in code to System.out.println( objectname) .. where objectname if the object your just created. What do you see? Why?