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.
4. Part 4: toString() -

**Part 1 – Explore 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/>

Java classes are grouped into *packages.*

Exercise

Find the **JButton** class which is in the **javax.swing** package

* 1. What is the purpose of this class
  2. How many “Constructors” does JButton have?

Find the **Arrays** in the **java.util** package

1. What is the purpose of this class
2. 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.

Look at the video here if you like a visual explanation of how to use it: https://www.youtube.com/watch?v=23tAK5zdQ9c

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 “Control” – as we’re going to use it to control /run things – it isnt’ a typical class with attributes.. etc.

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

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

\* Control: the purpose of this class.. etc

\* Author:

\* Date:

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

Write java code in your Control 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 and it’s outline is:

public String toString()

{

// your code goes here

}

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 object name if the object you just created. What do you see? Why?