

Title: Intro to Java lab

By the end of the lab, make sure to drop in your group Teams session to show your code to the lab assistant to get graded.

Note: If you are using Eclipse as your IDE, an **Eclipse** video is here if you prefer a visual explanation of how to use it:

<https://www.youtube.com/watch?v=23tAK5zdQ9c>

Open up Eclipse on your machine.

Eclipse will ask you to set the **workspace** (this is where your code will be stored).

1. Creating a Control java class to run your java code.

This part is setting up a class with the “main” method as discussed in 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 a java **class**: From the package, create a new class. Call it “Control” – as we’re going to use it to control /run things – it isn’t 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 simply print “helloworld” out to the console –

TU857/2 OO programming Labs

- You will need the “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).
- note: `System.out.println(" ");` is the java command for printing to the console.

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 helloworld printed out.

2. Create a Class - Vehicle

Create a new java **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 - including Owner name, registration number, maximum speed, colour, 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 and “registration” with a value.

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

You don’t “see” any result - as creating objects isn’t producing any visible output.

Add a **second** constructor to your Vehicle class – that sets up all attributes with initial values.

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

Use `System.out.println(whateveryouobjectnameis)` from the main `Control` class – to “print out” your object. What do you see? How useful is it as a representation of your object?

3. Add a method `toString()`

Add a method called “`toString`” to your `Vehicle` class. This method will return a `String` result that shows a readable version of the attributes.

– by concatenating* the values of the attributes of the class, formatted with text e.g. “This vehicle has owner name X , has a registration plate of A, is of colour Y etc”.

*Example of concatenation in java: `String summary = "This patient room number is name" + this.roomNumber;`

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

4. Set “recentCar” attribute

Add a boolean attribute called `recentCar`. It is not set by the constructor. Instead, add a method in your code `setRecentCar ()` that does the following:

If the registration is for a car more recent than 2002, it will set the attribute called “recentCar” to true. If the registration is older than 2002, set to false. If the registration is invalid (i.e. no year supplied), set `recentCar` to false. Note: registrations are of the form `YY(N)SSNNNN`, where `YY` is the year (02,03..etc)