Java Programming 2 – Eclipse intro

Before you start

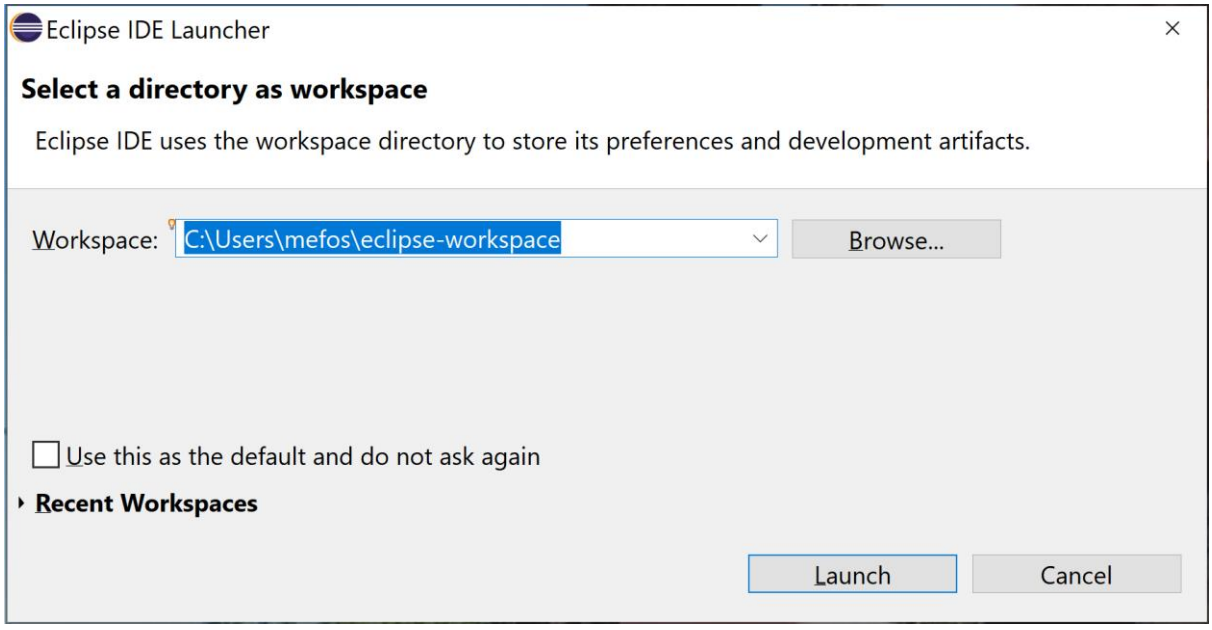
Download the file **FirstProject.zip** from the course Moodle site, in the same place as you downloaded this file.

If you haven't yet installed Eclipse on your computer, please follow the Moodle instructions for installing it – the exact instructions depend on the operating system of your computer.

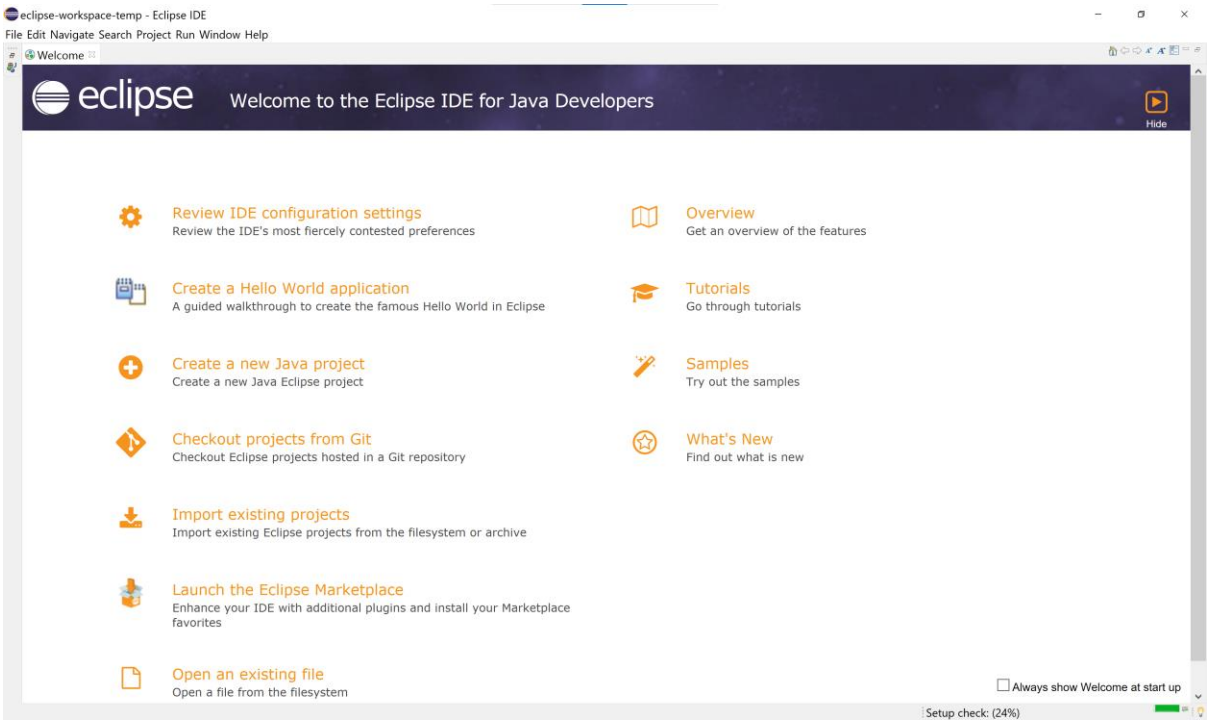
Using Eclipse

Eclipse is an open-source Integrated Development Environment (IDE) that will be used for coding, compiling, running, and debugging Java programs. Eclipse provides many more facilities than this, some of which you may use in later courses. It is a many-featured and versatile piece of software and can appear a little daunting for beginners. We will be using only a small subset of its capabilities, and a key objective of this document is to begin the process of familiarisation with these.

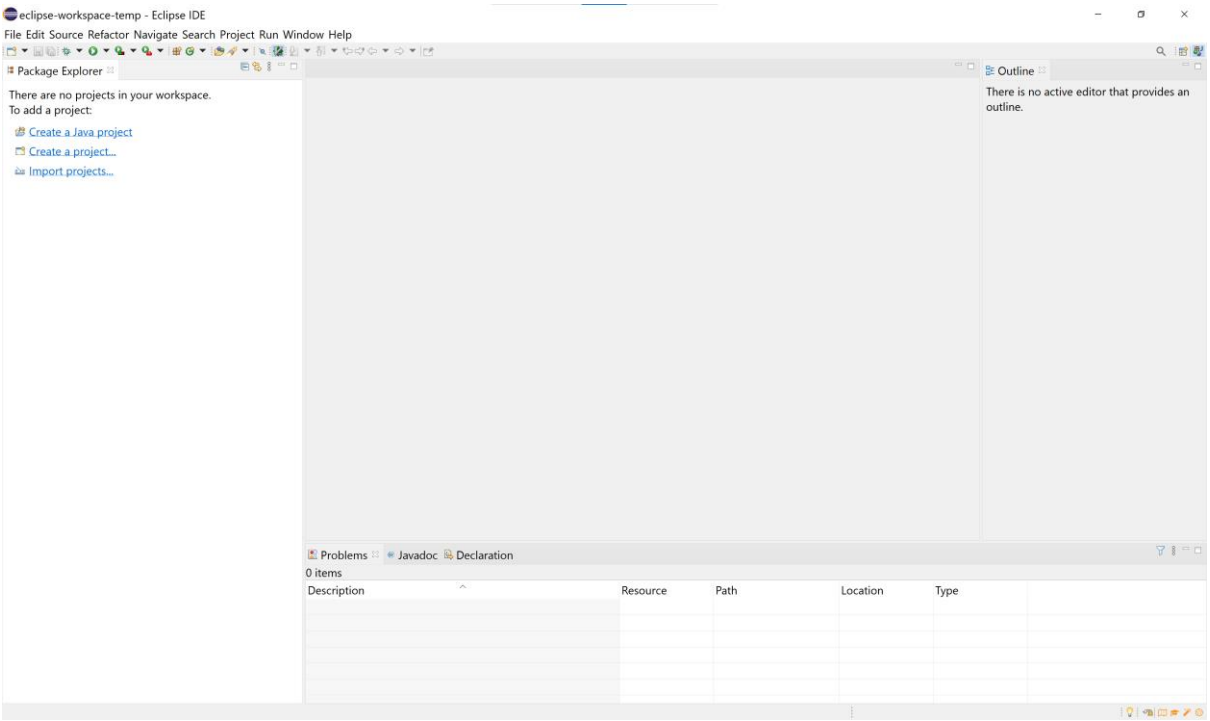
- Launch Eclipse on your computer. The process for doing this varies depending on your operating system.
- After it launches, you will first be asked to choose a **workspace folder** – this is the folder that will contain all of the files you are working on. On a Windows machine, your default workspace folder will be in **c:\Users\<username>\eclipse-workspace**; the path will be similar on other operating systems. You can leave it as this if you want, or you can change it to a different directory if you prefer, and you can tick the box to use this workspace as the default if you do not want to be asked again. Press **Launch** to launch Eclipse in the selected workspace.



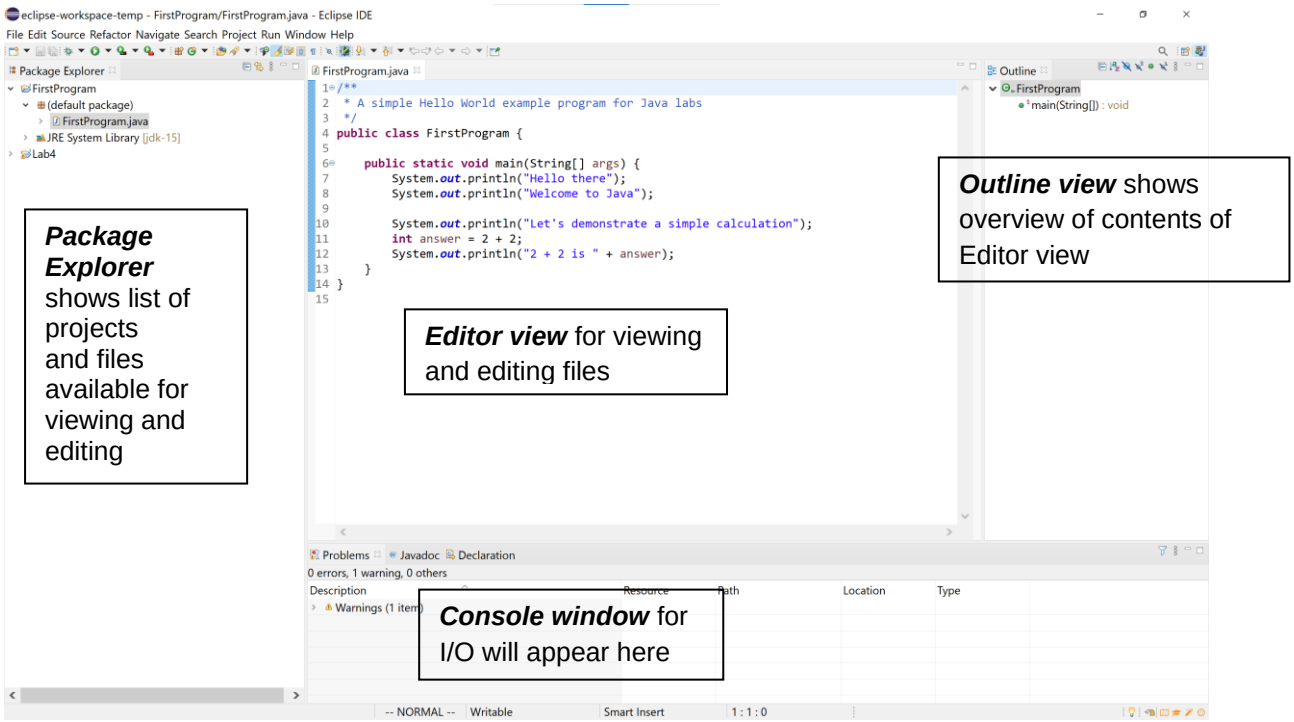
- Eclipse takes a few seconds to start up. You will see Eclipse's 'welcome' screen appear - see below. You can make sure that the “Always show Welcome at start up” box is un-ticked (it is un-ticked by default), to ensure that you go straight to the workspace when you open it later on.



- Clicking on the “Hide” arrow (arrow in box on top right hand side) will take you to the main IDE. The screenshot below shows the initial appearance of the Java window.



- Each Java program that you work on will form an Eclipse **project**, and this project has to be created. Typically in the rest of this course, some or all of the files that are required for a project will be downloaded, at least in skeleton form, when you download the lab zip file from Moodle. Each project will have its own folder, the name of the folder necessarily being the same as the name of the project. You will need to import the project(s) into your workspace so that you can modify and run them.
- In Eclipse, select **File** → **Import** … to launch the import wizard – this will be used to import the starter project into your workspace. Then select **General** → **Existing Projects into Workspace** and click **Next**.
- Choose **Select archive file** and then browse to the location where you downloaded FirstProject.zip. Select FirstProject.zip and click **Open**.
- You should now see a one project in the Projects window: **FirstProject**. Ensure that the checkbox beside this project is selected (e.g., by pressing **Select All** if it is not), and then press **Finish**.
- In the **Package Explorer** view (on the left of the Eclipse window), you should now see an icon representing the project. Clicking on the expander icon beside **FirstProject** reveals the contents of the project – namely, components of the Java Run-Time Environment (JRE) together with the *default package*, which contains the sole source code file for this program, i.e. FirstProgram.java. Expanding the default package icon will reveal this file. Then double click the file name to open the file in the central **Editor view**. You will see the source code for the 'Hello World' program, and in the **Outline view** you will see a summary (brief in this case) of the methods in this class.



- Now click on the **Editor view** to activate it. Whenever Eclipse detects an error in the java file displayed in the editor view, this is indicated by red underlines in the text, and red crosses to the left (and possibly also the right) of the view. Hovering the cursor over a marker throws up an error message – see below. This feature of Eclipse takes a bit of getting used to; it can be annoying as the compiler often does not give you time to finish typing a line before throwing up some spurious error indicator, but in general this can be a very useful feature once you get used to it.
- The program here is complete and contains no errors, hence Eclipse does not complain. But nonetheless, it has done its work behind the scenes, and you can run the program. To do this, first make sure that the **Editor view** is active (as indicated by a blue border). Then select the Menu option **Run → Run as → Java application**, or click the green 'Play' button in the toolbar. You should see the output from the program in the **Console view**, at the foot of the Eclipse window:

- Now introduce an error into the Java code. For example, delete one of the letters from the word `static`. You will see a red cross appear on the extreme left of the Editor view, and moving the cursor over this marker reveals a helpful error message. (Error messages are not always as immediately helpful as this.) Experiment with some further errors – for example, try deleting punctuation marks like semicolons or curly brackets to see what errors are indicated.
- Eclipse also supports auto-completion – in most cases, you can press **Control-Space** to complete a class name, method name, or field/variable/parameter name. Try declaring some more variables and/or methods and then calling then with auto-complete.
- You may have noticed that when the cursor hovers over an identifier (in the **Editor** view) information about that identifier is displayed.

Eclipse is very powerful, and you will discover a range of behaviours of Eclipse over the course of the semester – feel free to explore the menu options and settings.