



MACQUARIE
University

Department of Computing

ITEC625 Fundamentals of Computer Science
Workshop - Introduction

Learning outcomes

Following are this week's learning outcomes,

- a. introductions
- b. be ready to use the lab computers (setup accounts and iLearn)
- c. your first java program
- d. compiling and executing a java program
- e. introduction to eclipse
- f. importing a java project in eclipse
- g. adding a file to an eclipse project
- h. creating an eclipse project
- i.

1. Access your account

To log on to the lab machines, first make sure the machine is powered on. Enter your username (OneID) and password in the appropriate entry boxes. Please note that if you have already accessed the labs, your username and password are the same as in the previous semester, and if are accessing the labs for the first time, you should have the required information from enrolment/ orientation session. If you do not have this information, please ask the tutor to assist you.

Once you login to the computer, make sure you have access to ITEC625 homepage in iLearn. If not, please ask the tutor to assist you.

2. Writing your first Java program.

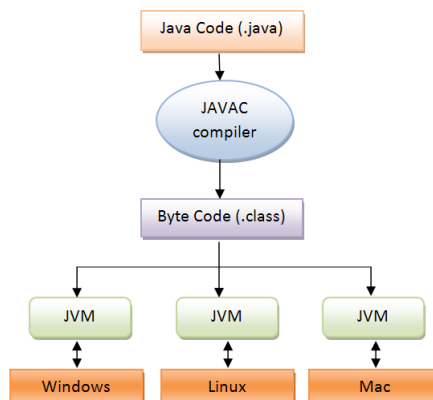
In a text editor (notepad or any other editor), type the following exactly as it appears,

```
public class HelloWorld {  
public static void main(String[] args) {  
System.out.println("Hello world");  
}  
}
```

Save this file as `HelloWorld.java`.

Next step is to *compile* and *execute* the program.

The following diagram describes the process of compiling a java source code (.java) into java bytecode (.class) and then executing the bytecode on the Java Virtual Machine (JVM).



Open Command Prompt/ PowerShell and go to the folder in which you saved the java file. Run the following command (or `ls` on a Mac/Linux) to list the contents of the directory:

```
dir
```

You should see the file (amongst others)

```
HelloWorld.java
```

Type the following command to compile `HelloWorld.java`.

```
javac HelloWorld.java
```

If everything is ok, you won't see any error message and if you run the `dir` command again, you will see the additional file

```
HelloWorld.class
```

If you *do* see an error message, take a look at the error message and fix the corresponding line. Ask your peers or tutor to help you if required.

Once the errors (if any) are fixed, type the following command to run/execute `HelloWorld.class`.

```
java HelloWorld
```

IMPORTANT: NO ".class" in the execution command.

3. Introduction to Eclipse

Your tutor will demonstrate the various aspects of Eclipse using a sample project.

The main aspects you should become familiar with, are,

- a. package explorer
 - Project
 - Package
 - * Class
 - Members
- b. console
- c. editor
- d. running a java program (a single project can have multiple runnable java programs, each called a *configuration*)

4. Importing and running Java projects in Eclipse

Follow the following instructions to import Java project contained in `eclipseDemo.zip` archive file.

- a. Download `eclipseDemo.zip`. **DO NOT UNZIP IT**.
- b. Open Eclipse IDE. A shortcut for it should be located on your desktop.
- c. If prompted, set your workspace (a location where all your projects will be saved). We suggest you use a dedicated folder in your network drive as the workspace.
- d. Click on **File → Import → General → Existing Projects into workspace (and NOT “Archive file”)**.
- e. On the next window, choose “Select archive file” option and browse for the archive file `eclipseDemo.zip` and choose Open.
- f. It should show a project `basicJavaProject`. Click on Finish.
- g. You should see a project in Eclipse in the left panel (Package Explorer).
- h. Double click on the `basicJavaProject` to reveal `src`. “`src`” is short for “source” (that is, source code).
- i. Double click on the `src` to reveal package “(default package)”.
- j. Double click on the (default package) to reveal three source files. Run them in the order below. Your tutor will help you on how to run these files.
 - (a) Program1
 - (b) Variables
 - (c) Expressions

5. Adding a new source file to an existing project.

In the project `basicJavaProject`, add a new source file by right-clicking on (default project) (in the left panel known as package explorer), choosing New and then Class. Name it `MyFirstClass` and **make sure to check** the `public static void main(String[] args)` inclusion option. Add a few lines of code from what you’ve learnt in the last three programs. Show this to your tutor.

6. Creating a new Eclipse Java Project

You can create a new java project in Eclipse by following the steps listed below,

- a. Click on File

- b. Choose `New`
- c. Choose `Java project` (not `Project`)
- d. Name the project `myFirstJavaProject` (project names begin with lowercases and are camelCased)

Then you can add a new file to the project as explained in the previous question.

Advanced questions

Some students might have done a fair bit of programming beforehand. These students should complete the following tasks.

1. acquaint themselves with the debugger feature of eclipse. create a java project and add a file Service containing the `main` method in it. Add a second method with the template:

```
/**
 * @param: n (assume n >= 0)
 * @return: sum of the first n powers of 2.
 * that is, 2^0 + 2^1 + 2^2 + ... 2^n.
 * Note that the value of this series is 2^(n+1) - 1
 * but you shouldn't use this.
 * Instead, compute the value iteratively using a loop
 */
public static int sumPowersOfTwo(int n) {
    return 0; //to be completed
}
```

Next, call the method in the `main` method by passing it a value 10. The value returned should be printed on the console and should be 2047. Try it for 15. It should be 65535. Try it for 25. Till when is the value valid? Why do you think it becomes invalid after a certain value of `n`?