OBJECT ORIENTED PROGRAMMING USING JAVA

Workshop Instructions

1 - Familiarization with the Java Environment

iSS

National University of Singapore

FAMILIARIZATION WITH THE JAVA ENVIRONMENT

Objectives

The objective of this workshop is to help you familiarise with the J2SE Environment, and set you up for the future workshops.
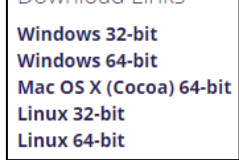
Setup

You will need to install Java Development Kit (JDK) if one does not already exist on your machine.
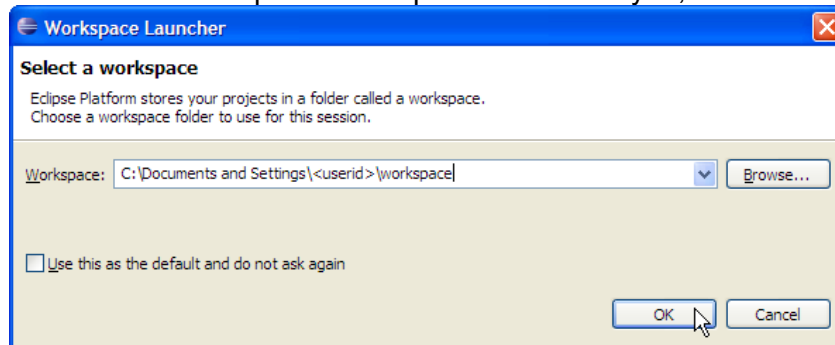
Exercise

Setting up Java Development Kit

1) Install JDK:
   a. Open http://www.oracle.com/technetwork/java/javase/downloads/index.html

   b. Click the DOWNLOAD button under JDK label
   c. On the next page, click "Accept License Agreement" and then download the latest installer for your operating system (Windows, Mac, Linux).
   d. Run the installer file to install JDK

Setting up your working environment

2) Install Eclipse:
   a. Open http://www.eclipse.org/downloads/
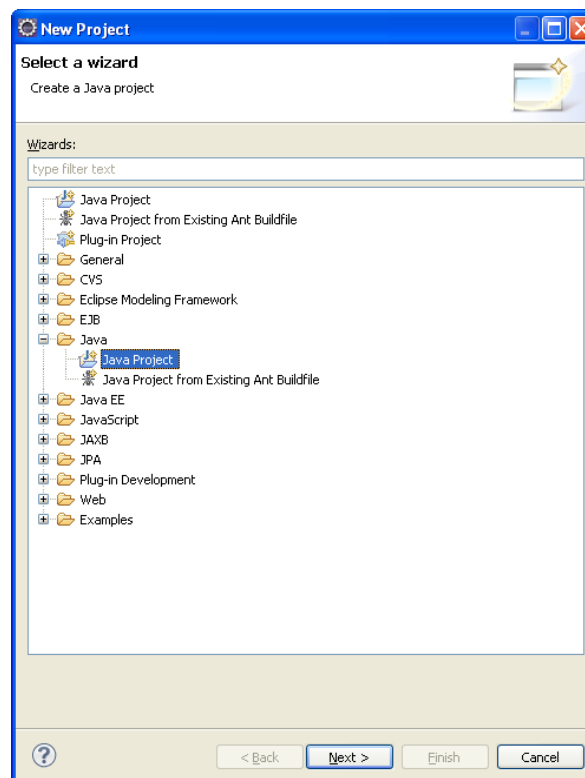   b. Click the link "Eclipse IDE for Java EE Developers"
   c. Under Download Links, download the Eclipse file for your operating system (Windows, Mac, Linux).
   d. Unzip the Eclipse file to any empty folder in your computer. Suggest to unzip it to c:\eclipse

3) Go to your Eclipse folder and run **eclipse.exe**. You will be asked to choose a workspace to store your Java project codes.
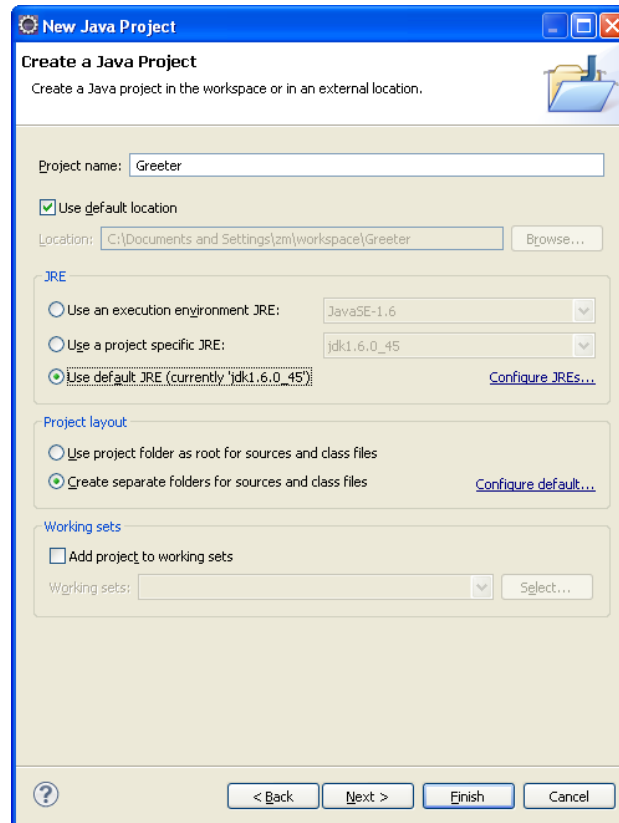4) Proceed to choose a workspace that is personalized for you, for instance as below:

Getting started with eclipse - Creating a Project

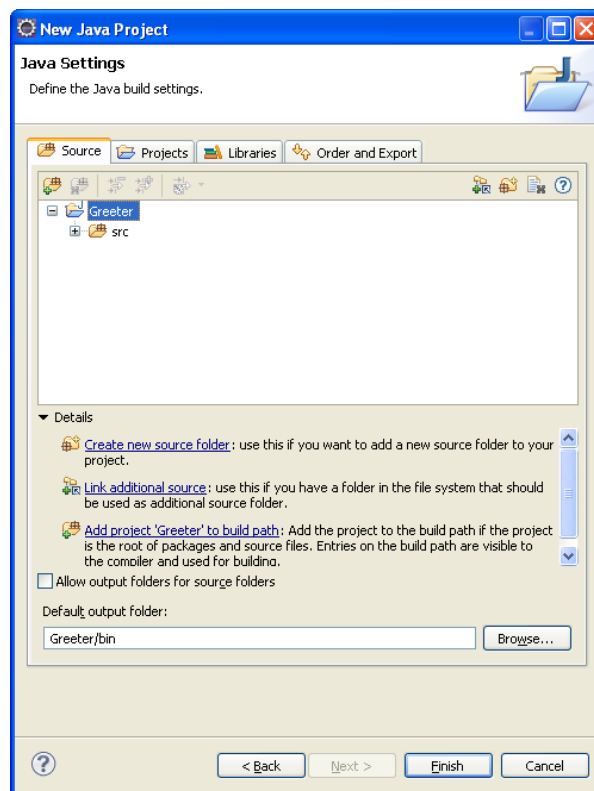5) Creating a Java Project
  a) There are three ways to create a Java project:
     i)   Select File > New > Project,
     ii) Select the arrow of the button in the upper left of the toolbar. Select Project, or
     iii) Right click on the Package Explorer view in the Java Perspective, and select New > Project.
     iv) Click on the icon on the toolbar.
  b) Select Java project and click Next.

c) Give the project a name. If you would like to create src and bin folders for your java and class files respectively select the Create separate source and output folders radio button under the Project layout area of the New Java Project dialog. The default source folder is src/ and the default output folder is bin/. You can change these defaults by clicking on the Configure default... link. Click the Next button.
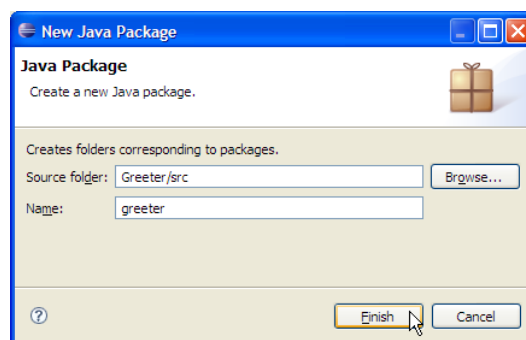

d) The next screen has four tabs: Source, Projects, Libraries, Order and Export. The **source** tab sets the source and output folders for Java files and class files. The **project** tab allows you to connect your new Java project to any other Java projects that are needed on the new Java project's build path. The **libraries** tab allows you to attach jar files of and Java libraries that you may need. The **order and export** tab determines the build order and how a project can be exported. Make any changes needed here, and then click the Finish button. If you are not in the Java perspective, Eclipse will ask if you would like to switch to the Java perspective. The project will show up in the Package Explorer view.

Getting started with eclipse - Creating a Java Package

6) There are four ways to create a Java Package. First, select the project you wish to create a package in.
   a) Select File > New > Package,
   b) Select the arrow of the button in the upper left of the toolbar. Select Package,
   c) Right click on a project in the Package Explorer view in the Java Perspective, and select New > Package, or
   d) Click on the icon in the toolbar.

7) Check to make sure that you are creating the package in the proper project and source folder. Give the package a name. Click Finish.
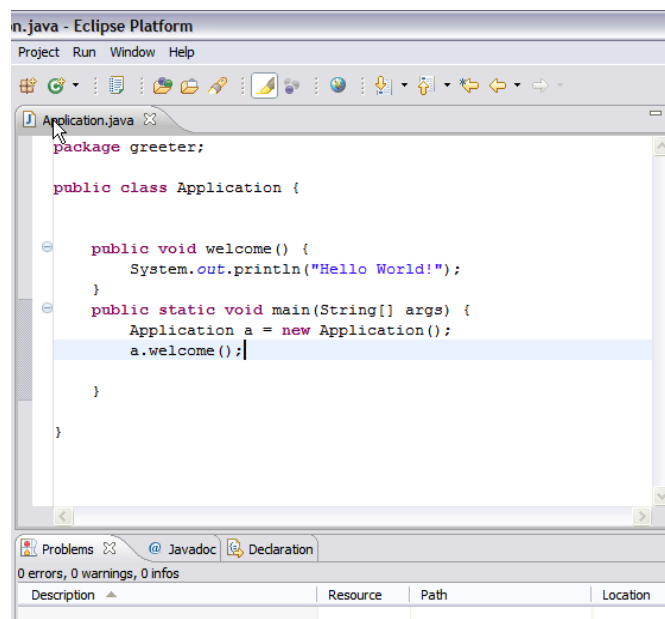
Getting started with eclipse - Creating a Java Class

8) There are four ways to create a Java Class. First, select the package you wish to create the class in.
    a) Select File > New > Class
    b) Select the arrow of the button in the upper left of the toolbar. Select Class,
    c) Right click on a package in the Package Explorer view in the Java Perspective, and select New > Class, or
    d) Click on the arrow of the icon in the toolbar. Select Class.

9) Check to make sure that you are creating the Class in the proper package. Give the class a name.
    a) Use the Browse button to search for a super class if your new class should inherit from another class.
    b) Use the Add button to add interfaces the new class should extend.
    c) Check the public static void main(String [] args) checkbox if the class you are creating starts the application.
    d) Check the Inherit abstract methods checkbox if you would like to inherit abstract methods from a super class or interface.

10) Click Finish. The new class will be open in the editor.

Complete the code

    11) Complete the code for class Application to print a welcome message as below

Getting started with eclipse - Running the Program
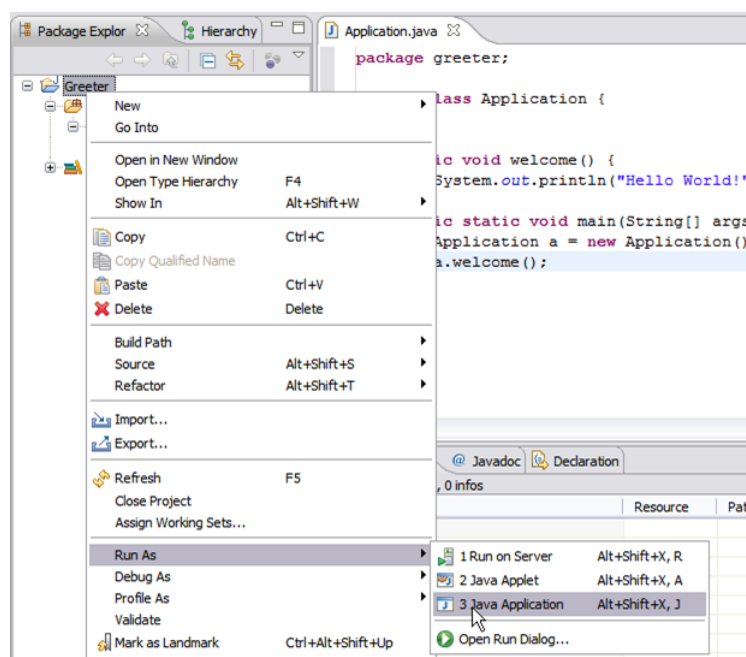
    12) There are several ways to run a program.
        a) If the program has never been run before, you need to create run configuration for the program.
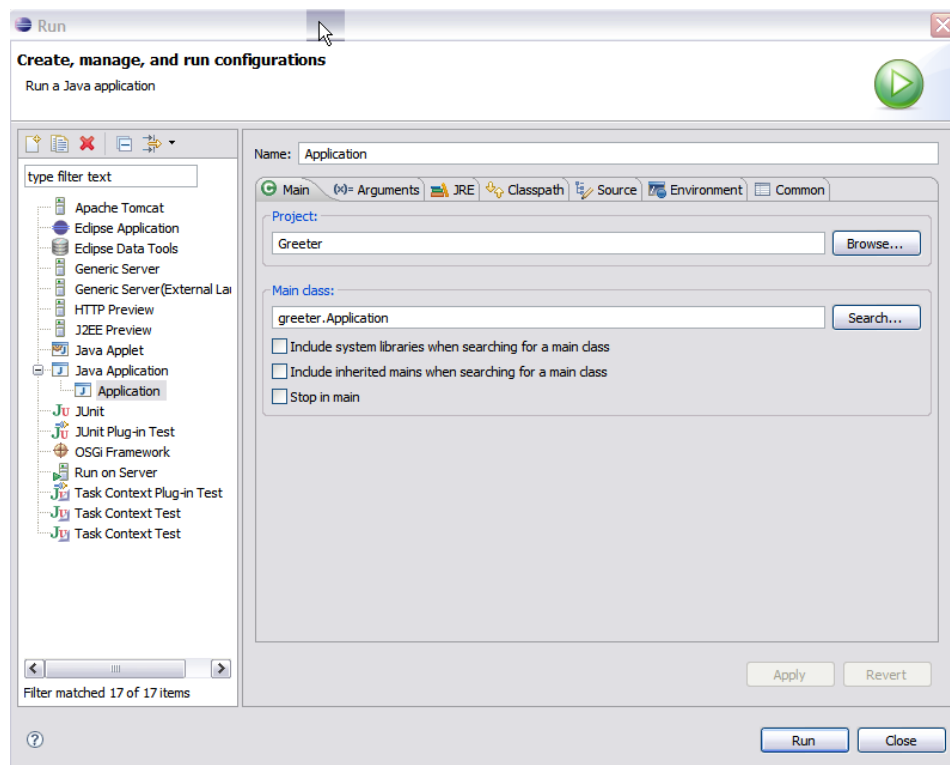   b) Select Run > Run Configurations..
        c) Click the arrow on the icon in the tool bar and select Run Configurations..
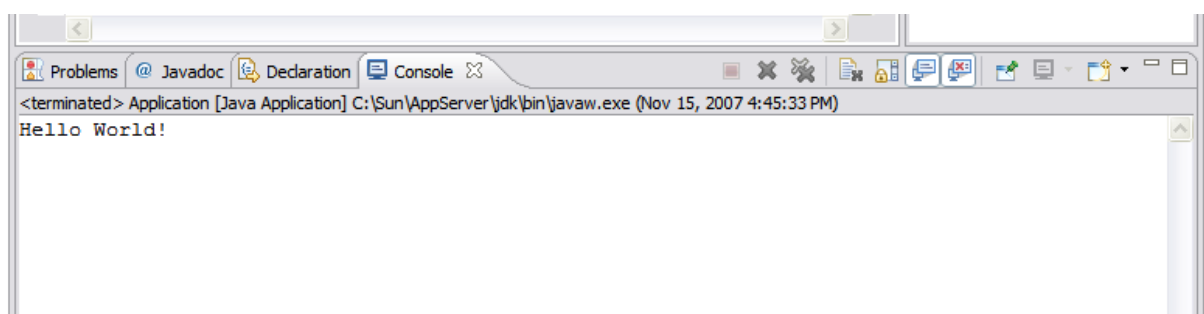   d) Right click on the project you wish to run. Select Run As > Java Application or Run As > Run Configurations..

13) Select Java Application, then click the New button at the bottom left of the screen. This creates a new run configuration. Give the setting a name. If the project is not specified, you can Browse for it. To find the main class in the project, select the Search button.

14) If there are any command line arguments, enter them in the Program arguments text box under the Arguments tab.

15) You can add jar files to the classpath under the Classpath tab.

16) Click Run to run the program.

17) If you have already created a run configuration, you can select the configuration by name by selecting the arrow on the [icon] icon in the tool bar. Just pressing the green button will run the last configuration launched by Eclipse.

Congratulations! You successfully run your first Java Program.

ISS
National University of Singapore