

Core Java 8 and Development Tools

Lesson 02 : Eclipse 4.4 (Luna) as
an IDE

Lesson Objectives

- After completing this lesson, participants will be able to:
 - Understand fundamentals of working with Eclipse
 - Creating and Managing Java Projects through Eclipse IDE
 - Use different features of Eclipse to develop rapid applications



Installing Eclipse 4.4 (Luna)

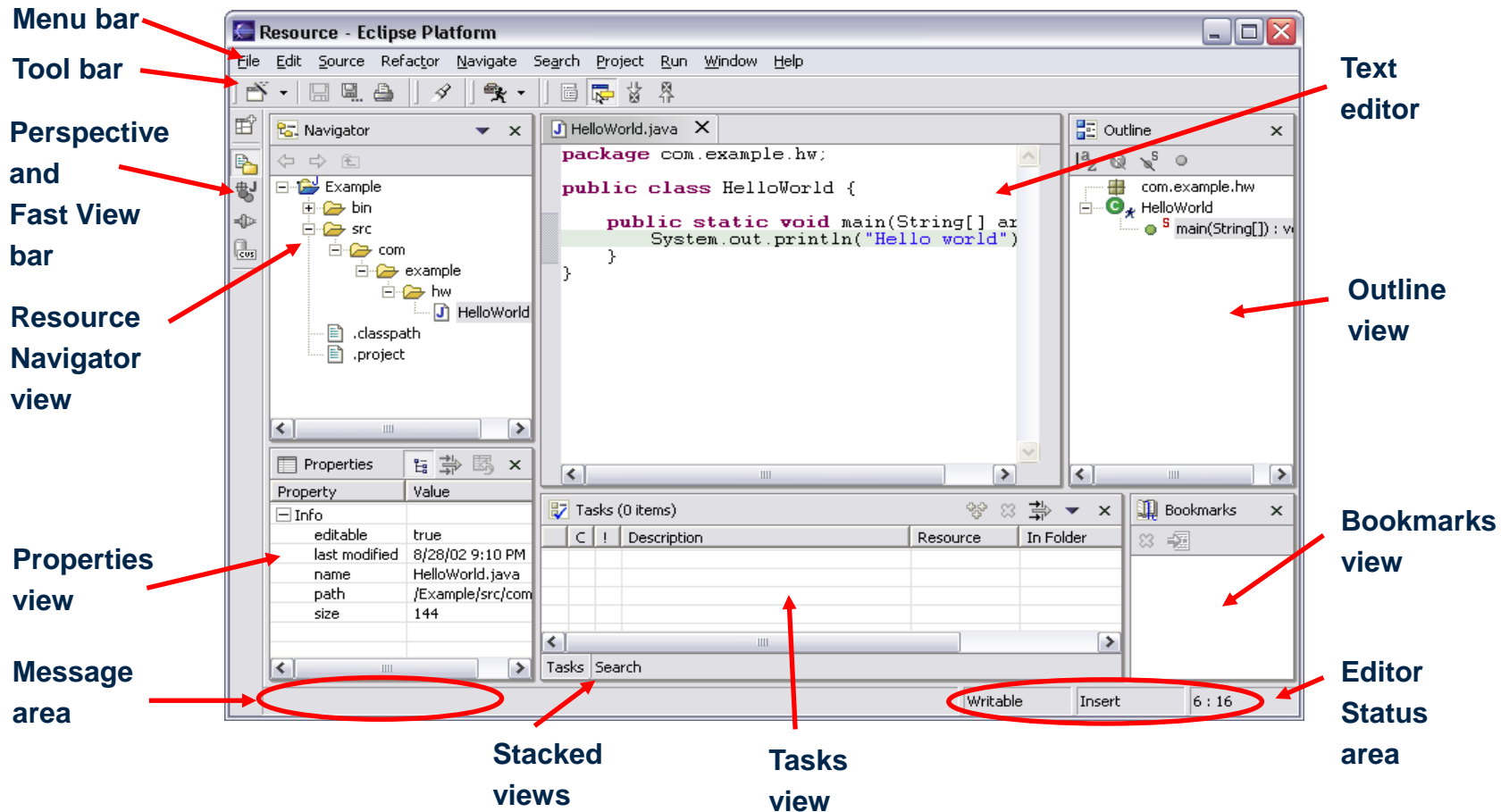
- You need to follow the given steps to install Eclipse 4.4:
 - Download Eclipse-SDK zip file from <https://eclipse.org/downloads/>
 - Unpack the Eclipse SDK into the target directory
 - For example: c:\eclipse4.4
 - To start Eclipse, go to the eclipse subdirectory of the folder in which you extracted the zip file
(for example: c:\eclipse4.4\eclipse) and run eclipse.exe

Integrated Development Environment

- IDE is an application or set of tools that allows a programmer to write, compile, edit, and in some cases test and debug within an integrated, interactive environment
- IDE combines:
 - Editor
 - Compiler
 - Runtime environment
 - debugger

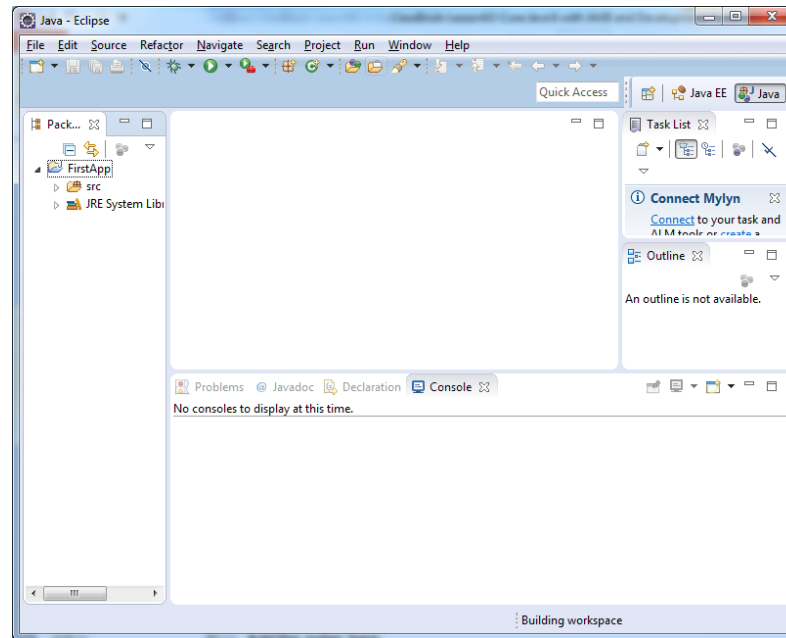


Workbench Terminology



The Workbench

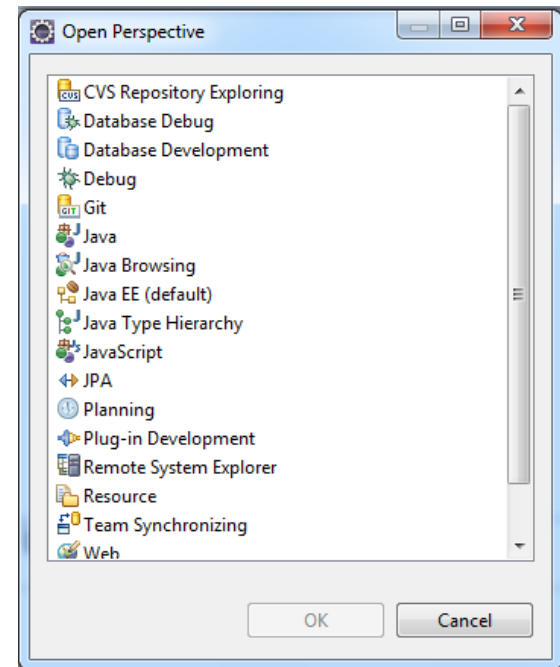
- The term “Workbench” refers to the desktop development environment
- It allows you to select the Workspace
- A Workbench consists of the following:
 - perspectives
 - views
 - editors



The Workbench

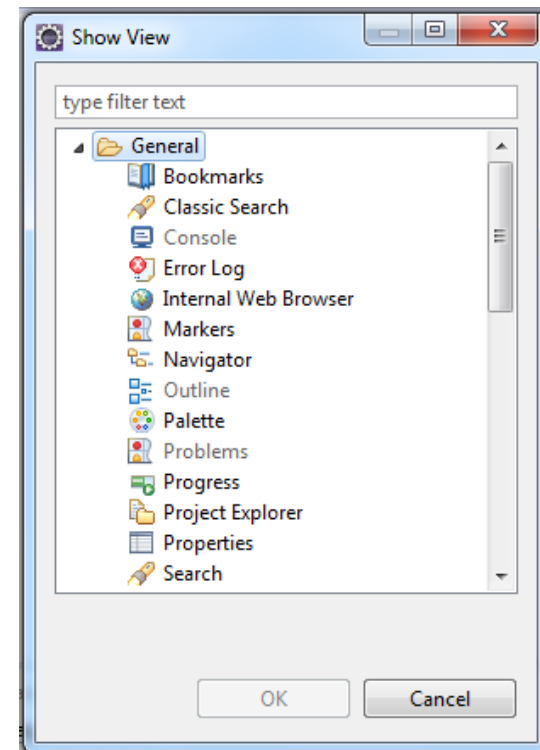
■ Perspective:

- A perspective defines the initial set and layout of views in the Workbench window
 - Workbench offers one or more Perspectives
 - A perspective contains editors and views, such as the Navigator
 - By default the **Java perspective** is selected
 - The title bar indicates which perspective is open



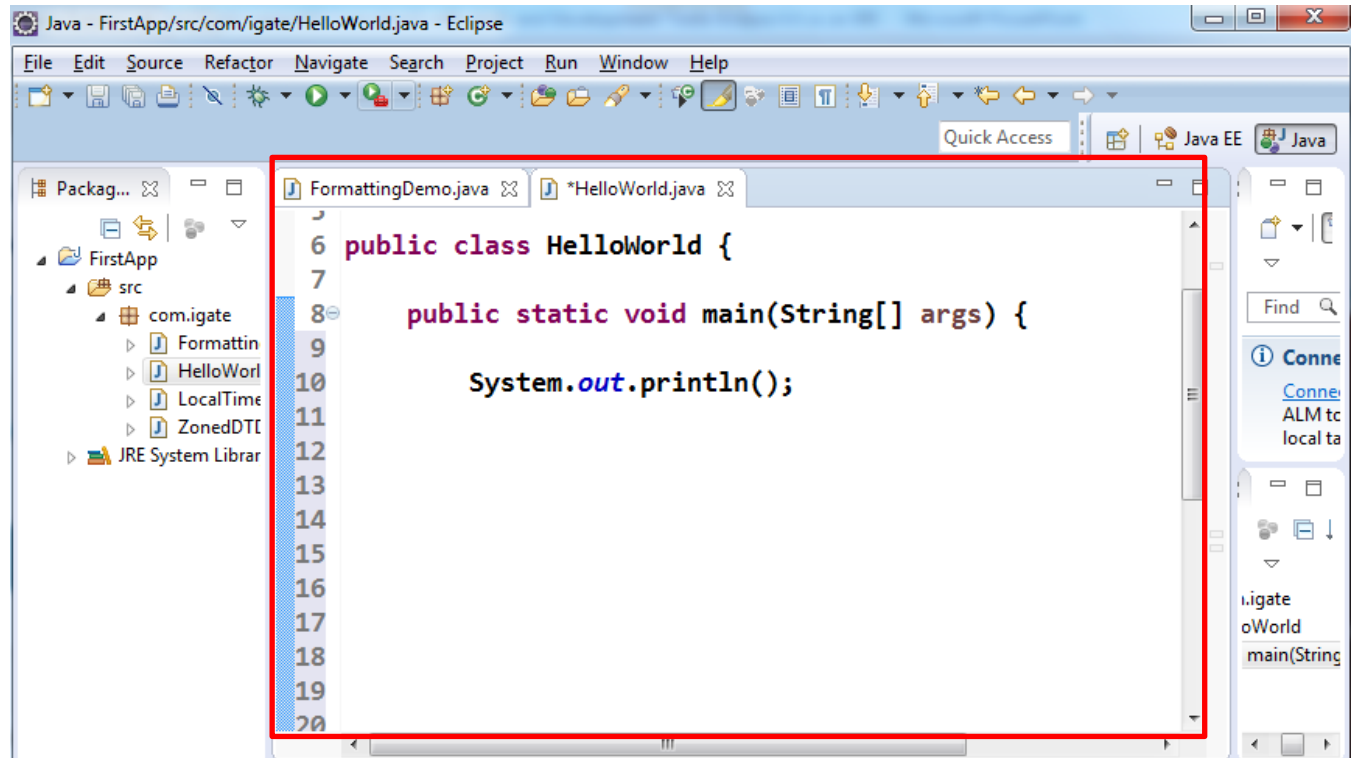
The Workbench

- View:
 - It is the visual component within the Workbench
 - It is used to navigate a hierarchy of information or display properties for the active editor



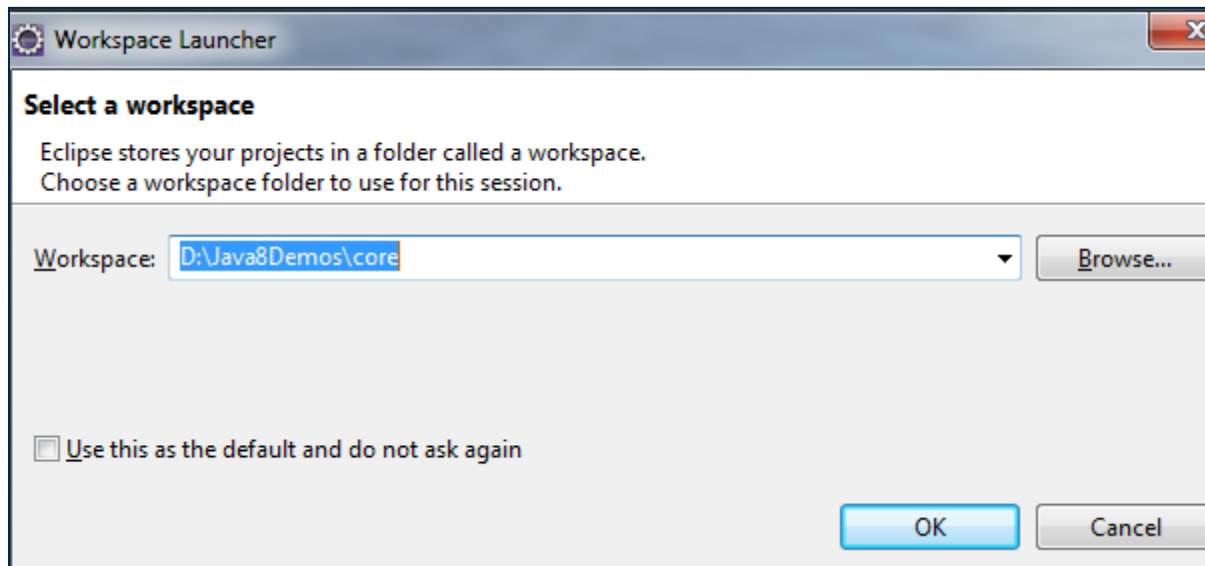
The Workbench

- Editor:
 - It is the visual component within the Workbench
 - It is used to edit or browse a resource



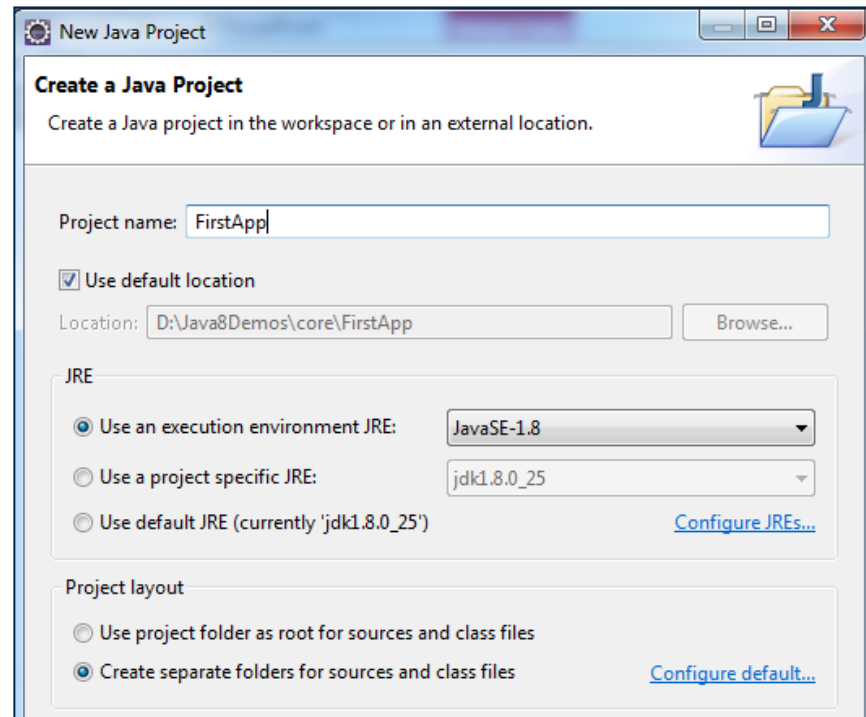
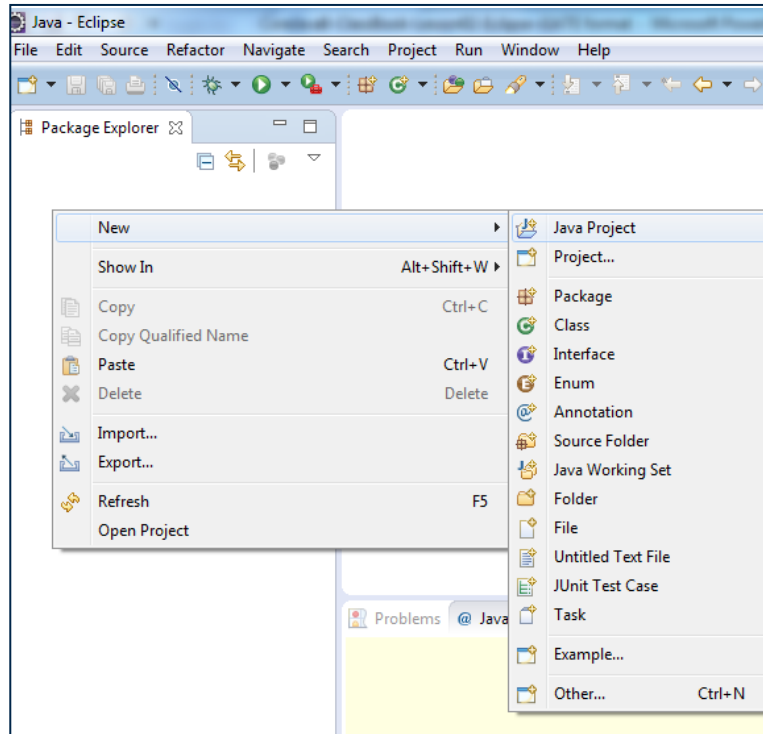
Create Workspace

- You need to follow the given steps to create a workspace:
 - Start up Eclipse
 - Supply a path to a new folder which will serve as your workspace
 - The workspace is a folder which Eclipse uses to store your source code



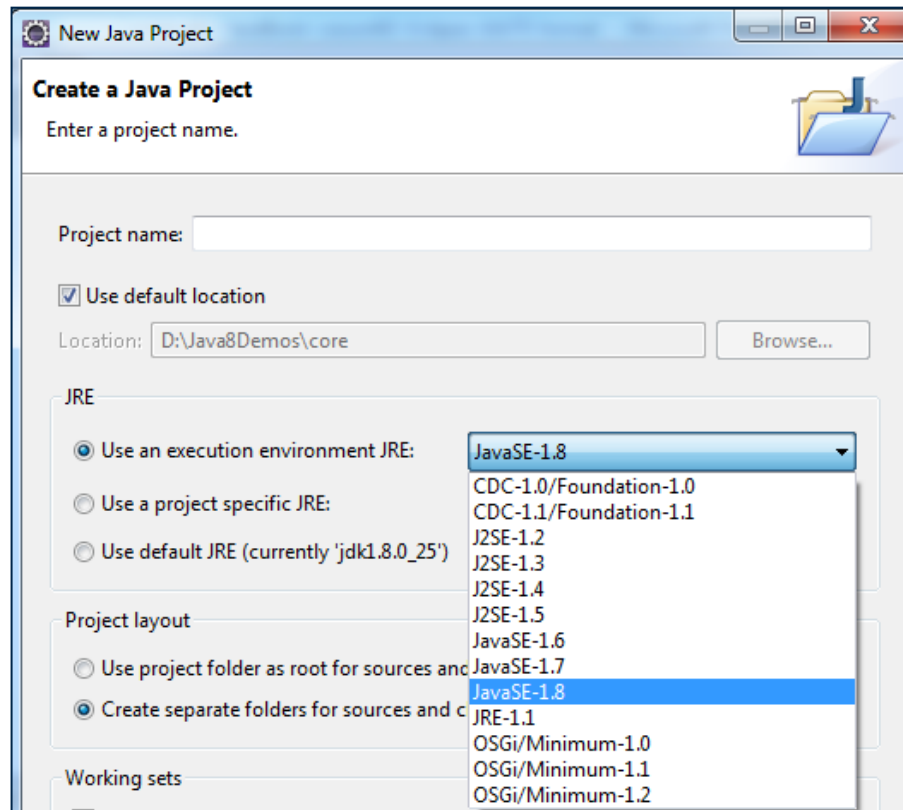
Create a Java Project

- Right-click the Package Explorer panel, and select New-JavaProject.
- Select Java project and provide a Project Name.



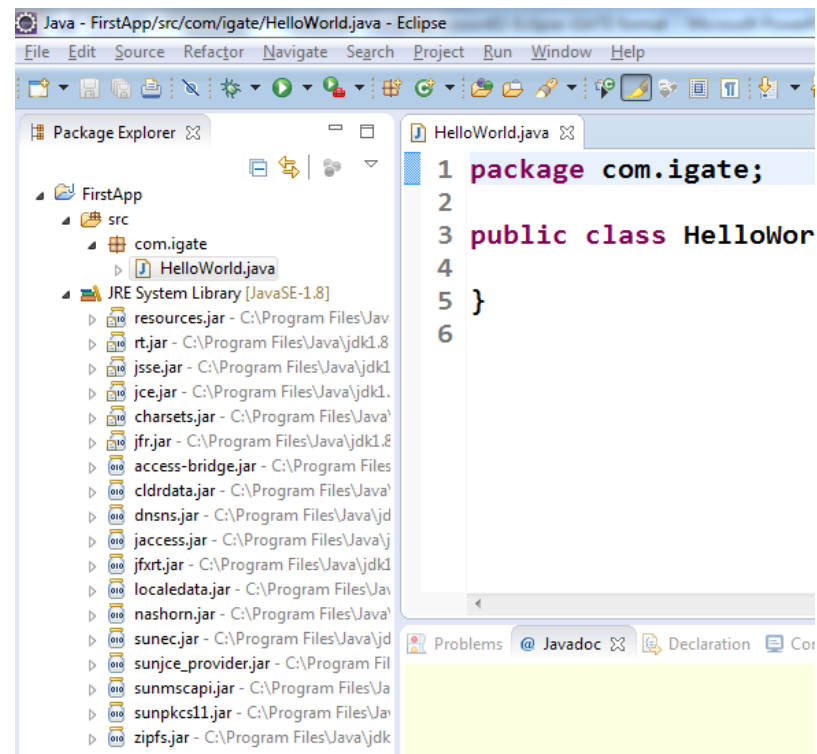
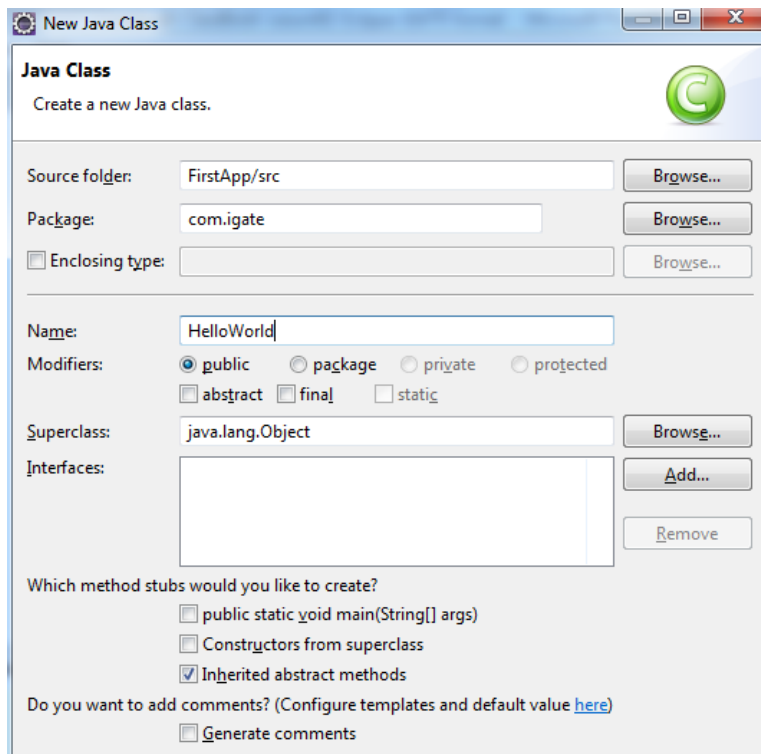
Select the JRE

- In order to develop code compliant with Java SE 8, you will need a JavaSE-1.8 Java Runtime Environment (JRE)



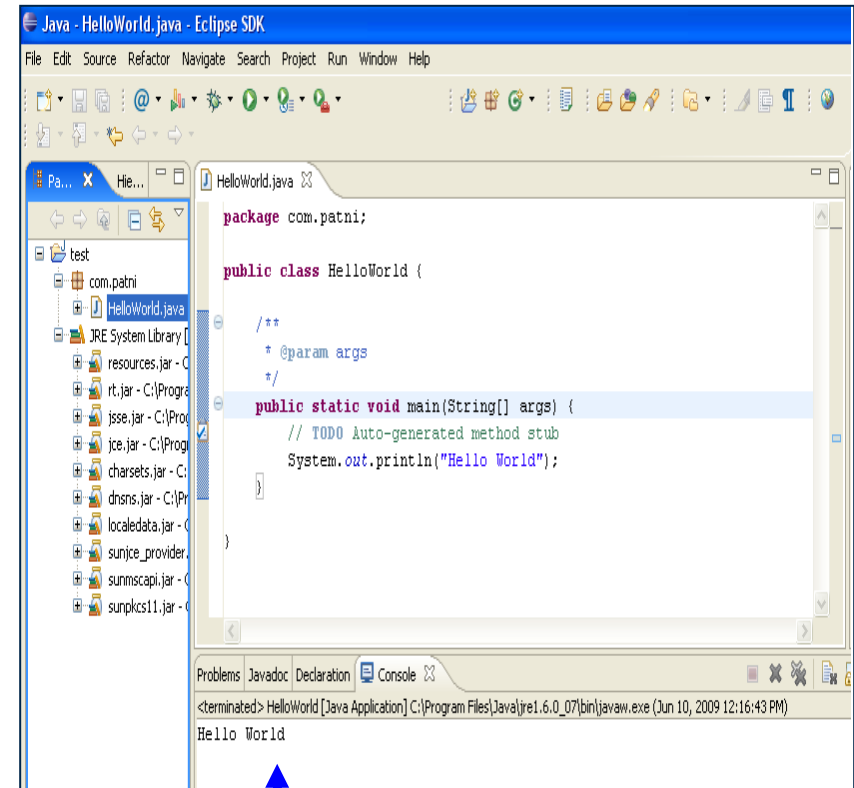
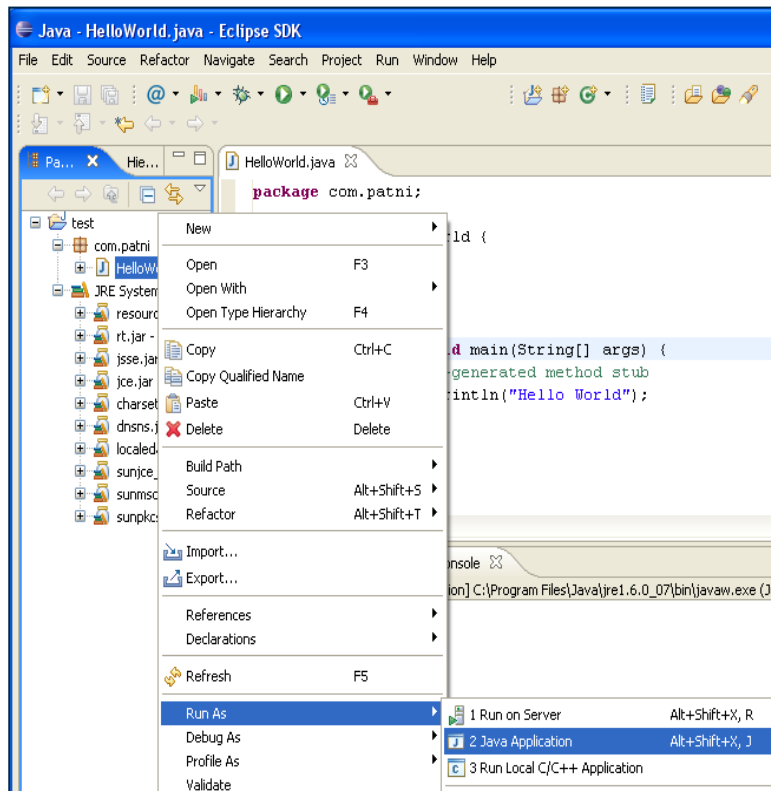
My first Java Program – Hello World

- Right-click on the project and select "New->Class" Type in your Program code



Executing Hello World Program

- Right-click the program and select Run As-Java Application.



Demo

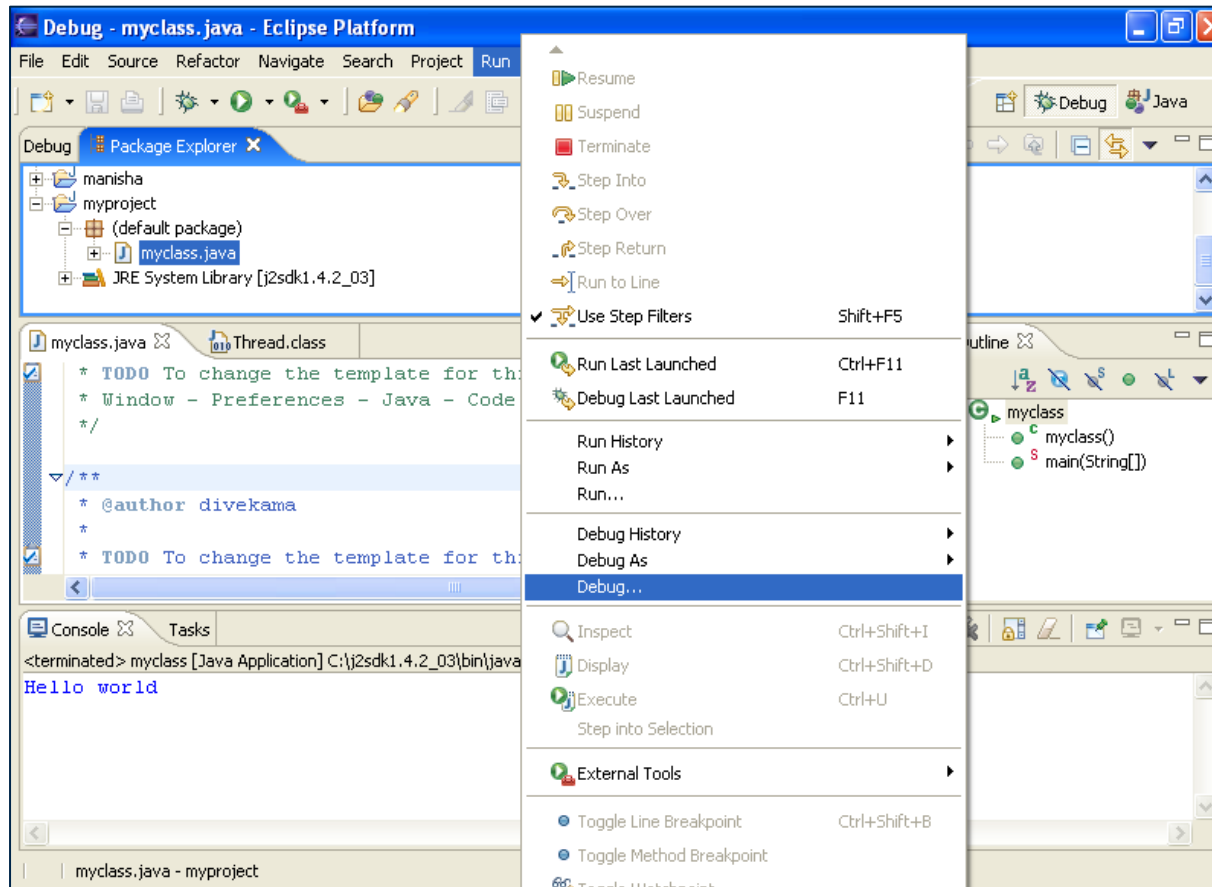
- HelloWorld Program using Eclipse IDE



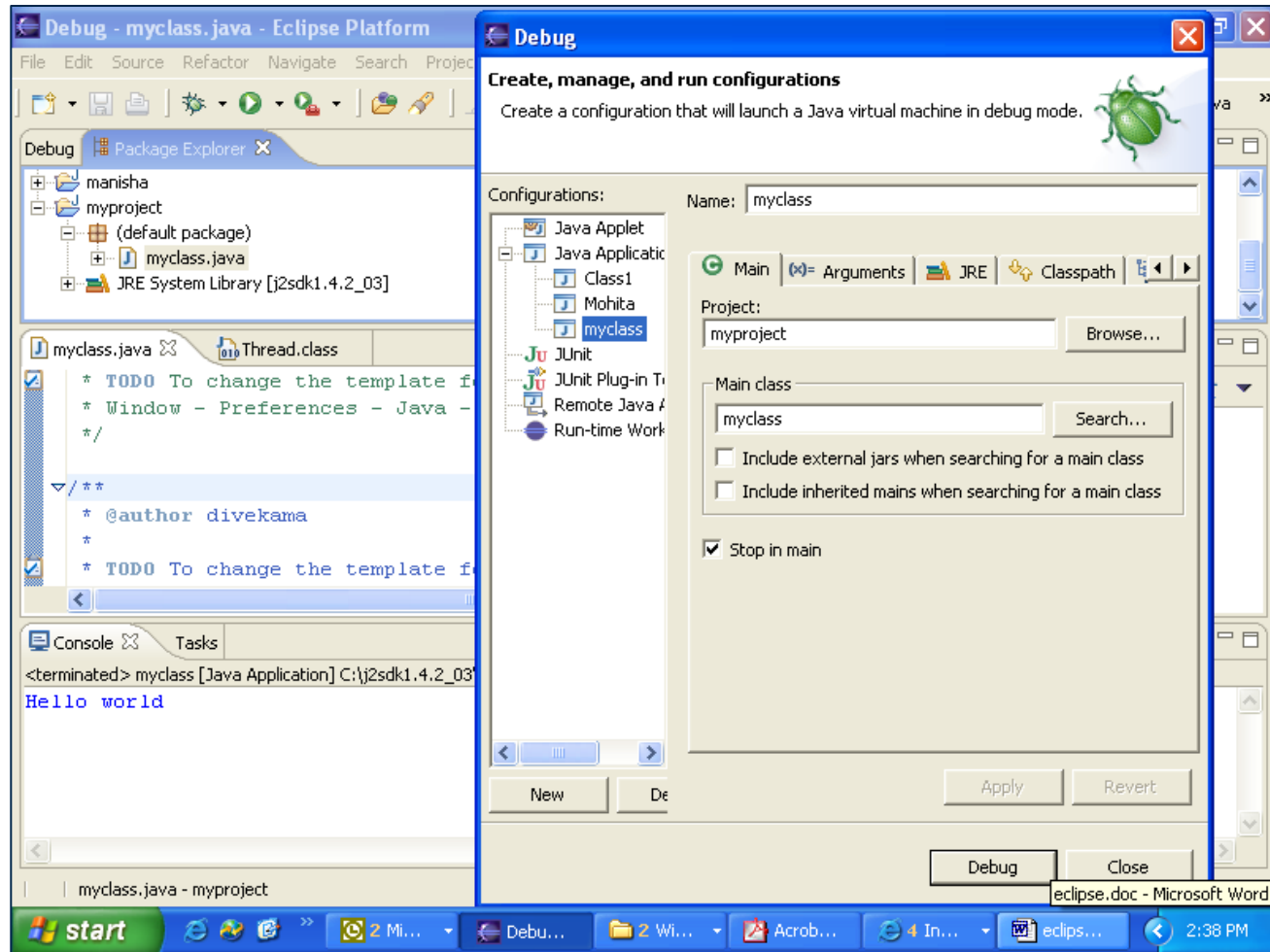
Debugging your Java Program using Eclipse

- The Java Development Toolkit (JDT) includes a debugger that enables you to detect and diagnose errors in your programs running either locally or remotely
- The debugger allows you to control the execution of your program by employing the following:
 - setting breakpoints, suspending launched programs, stepping through your code, and examining the contents of variables

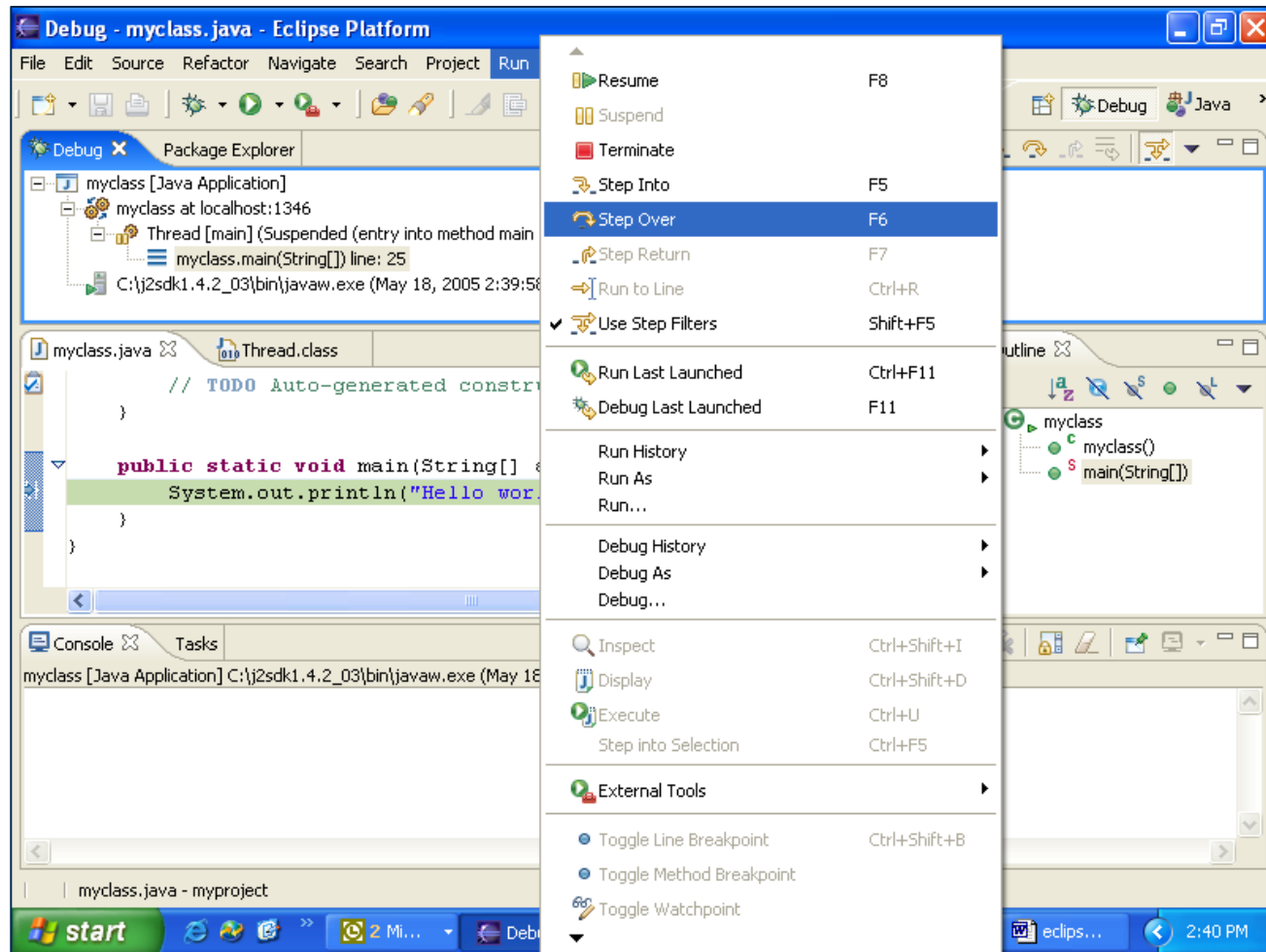
Debugging your Java Program using Eclipse



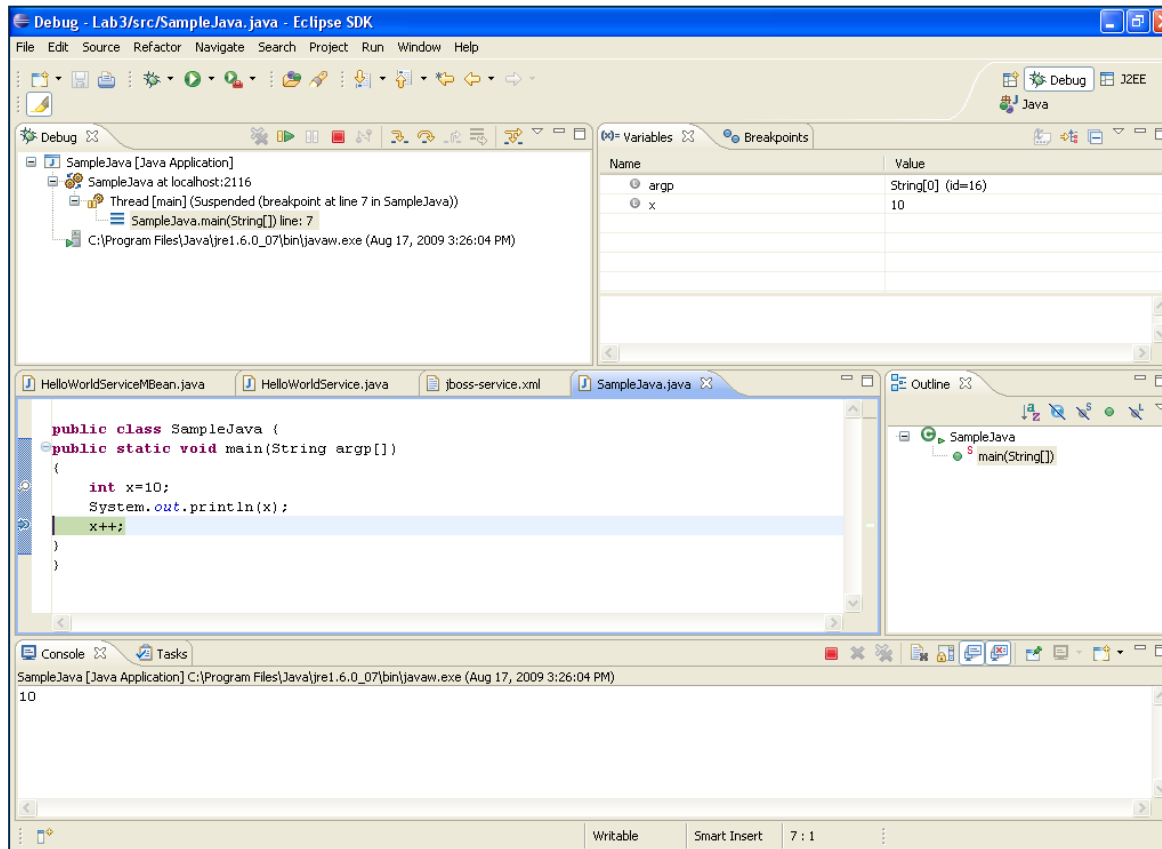
Specifying Debugging options



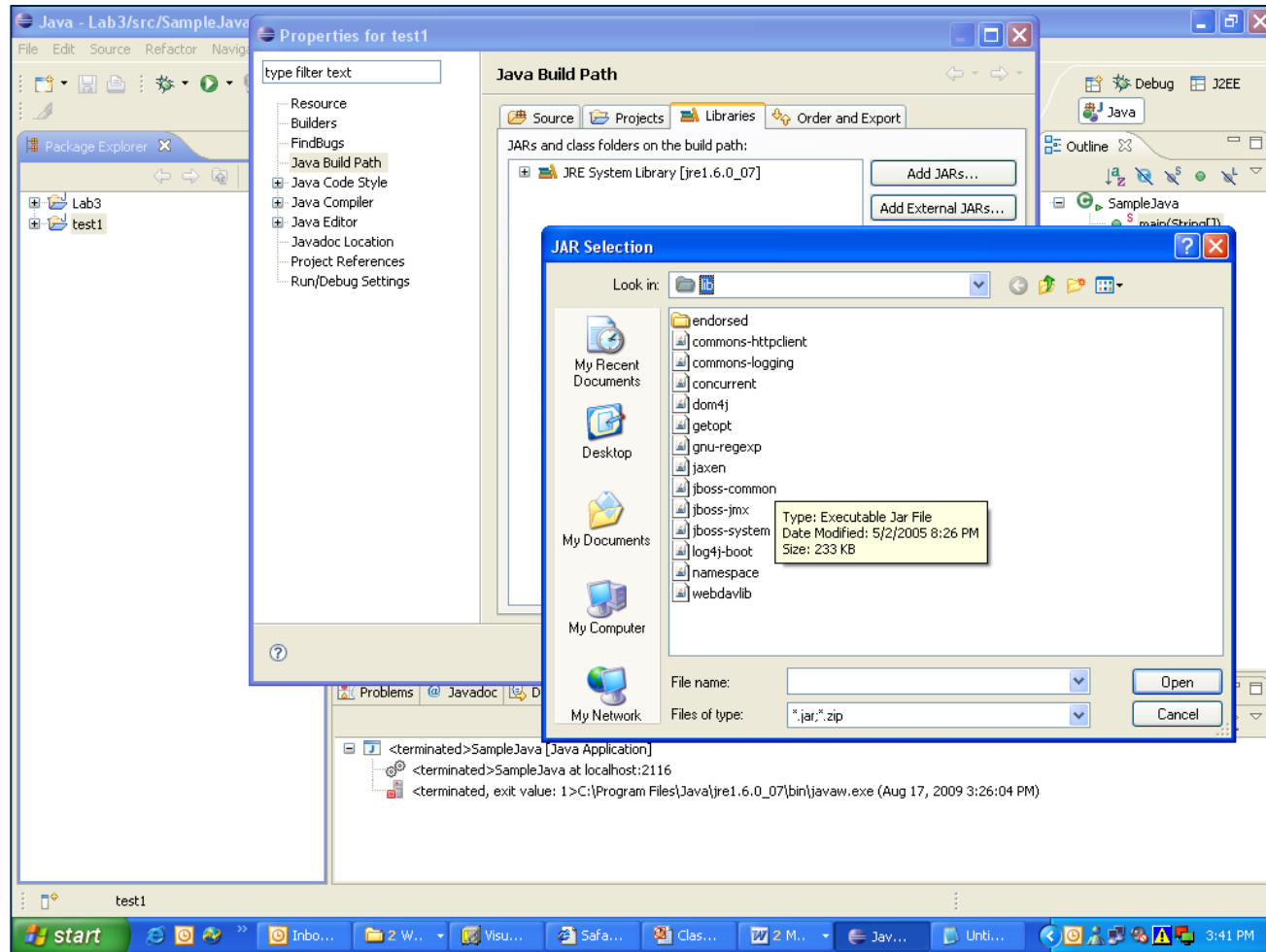
Debugging a Java Program



Debugging a Java Program



Adding external jar file



Verifying / Changing JRE Installation

- Changing the JRE is a common need while working with Eclipse which can be achieved as follows:
 - Select the menu item Window → Preferences to open the workbench preferences
 - Select Java → Installed JREs in tree pane on the left, to display the Installed Java Runtime Environments preference page
 - To add a new JRE, click the Add button, and select the new JRE home directory
 - Change the appropriate compiler.
 - Select Java → Compiler and select the appropriate compiler

Jar File Creation

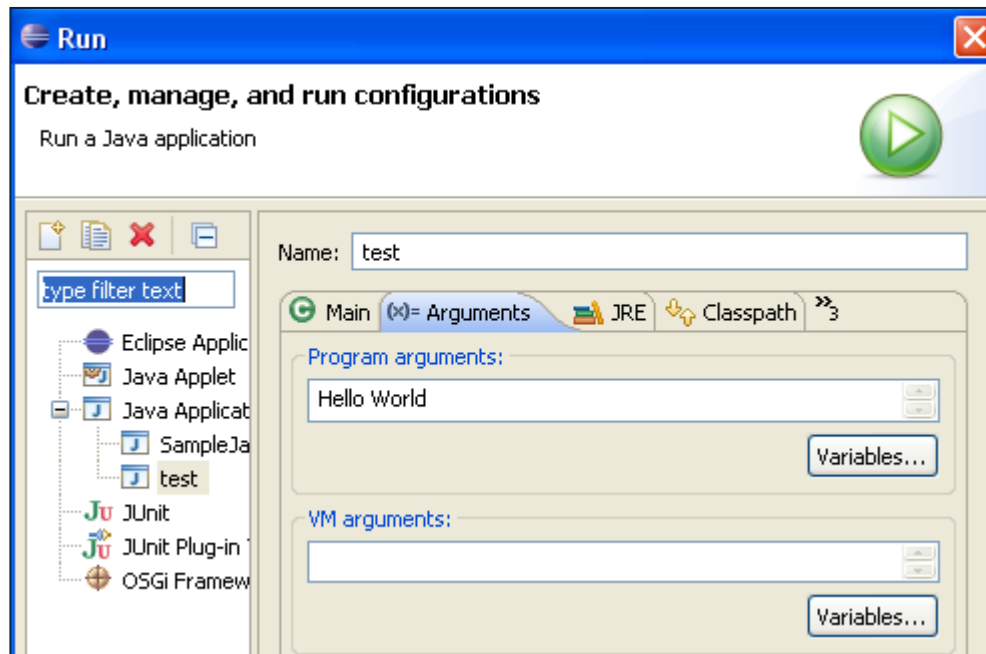
- In the Package Explorer, you can optionally pre-select one or more Java elements to export
 - Select Export from either the Context menu or from the File menu
 - Expand the Java node, and select JAR file, and click Next
 - On the JAR File Specification page, select the resources that you want to export
 - Specify a name to the JAR file
 - Click Finish to create the JAR file

Class path Setting

- Classpath variables allow you to avoid references to the location of a JAR file on your local file system
- Classpath variables can be used in a Java Build Path to avoid a reference to the local file system
- The value of such variables is configured at the following path:
 - **Window → Preferences → Java → Build Path → Classpath Variables**

Passing Command Line Arguments

- Command line arguments can be passed to the program in the following ways:
 - Select **Run** → **Open Run** dialog → **Arguments** tab



Import a Project

- To import an existing project to the workspace:
 - Go to File ☐ Import
 - Select Existing Projects into Workspace option
 - Select the radio button next to Select archive file, and click the Browse button
 - Find the archive file on your hard disk and click Open to select
 - If you have selected an archive file containing an entire Eclipse project, then the project name will appear in the box below, that is already checked
 - Click Finish to perform the import

Build options

- By default, builds are performed automatically when you save resources
- Two types of Build are available, namely:
 - **Auto Build:** By selecting **Project** → **Build automatically**
 - **Manual build:** By deselecting **Project** → **Build automatically**
 - It is desirable in cases where you know building should wait until you finish a large set of changes
- To build all the resources from the scratch you have to select **Project** → **Clean**

General Tips and Tricks

- Creating Getters and Setters:
 - To create getter and setter methods for a field:
 - Select the field's declaration
 - Invoke Source → Generate Getter and Setter
- Content assist:
 - Content assist provides you with a list of suggested completions for partially entered strings
 - In the Java editor, press CTRL+SPACE or invoke Edit → Content Assist

General Tips and Tricks

- Source menu contains a lot of options which can be used during code generation:
 - **Code Comments:** You can quickly add and remove comments in a Java expression
 - **Import Statements:** You can use it to clean up unresolved references, add import statements, and remove unneeded ones
 - **Method Stubs:** You can create a stub for an existing method by dragging it from one class to another

General Tips and Tricks

- **Try / Catch statements:** You can create Try / Catch block for expression by Source → Surround with try/catch
- **Javadoc Comments:** You can generate Javadoc comments for classes and methods with Source → Add Javadoc Comment
- **Superclass constructor:** Add the superclass constructors with Source → Add Constructor from Superclass

Using Java documentation

- For new developers, to quickly get familiar with the Java API, Java provides API documentation.
- The documentation also provides description and examples for all methods of each class.
- It can be downloaded from <http://docs.oracle.com/javase/8/docs/api/> for offline access.
- To see Java documentation for any class or method, eclipse provides “javadoc” view.
- To enable this view, select Windows ☐ Show View ☐ Javadoc.
- You can also view the javadoc contents in HTML format by using shortcut key “Shift + F2”.

Lab

- Lab 1: Working with Java & Eclipse



Summary

- In this lesson, you have learnt:
 - The method to install Eclipse
 - Process to create a Java Project with Eclipse
 - Various useful features of Eclipse



Review Question

- Question 1: Which of the following are true with Eclipse 4.4?
 - **Option 1:** A Java Project in Eclipse has got a Java builder that can incrementally compile Java source files as they are changed
 - **Option 2:** A workspace can have one project only
 - **Option 3:** The source and class files can be kept in different folders
- Question 2: To build all resources, even those that have not changed since the last build, you have to select the following option:
 - **Option1:** Project → Build Project
 - **Option2:** Project → Build All
 - **Option3:** Project → Clean

