Core Java 8 and Development Tools

Lesson 02 : Eclipse 4.4 (Luna) as an IDE

Capgemini

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

© 2017 Capgemini. All rights reserved.

2.1: Installation and Setting up Eclipse
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

© 2017 Capgemini. All rights reserved.

2.2 : Introduction to Eclipse IDE
Integrated Development Environment

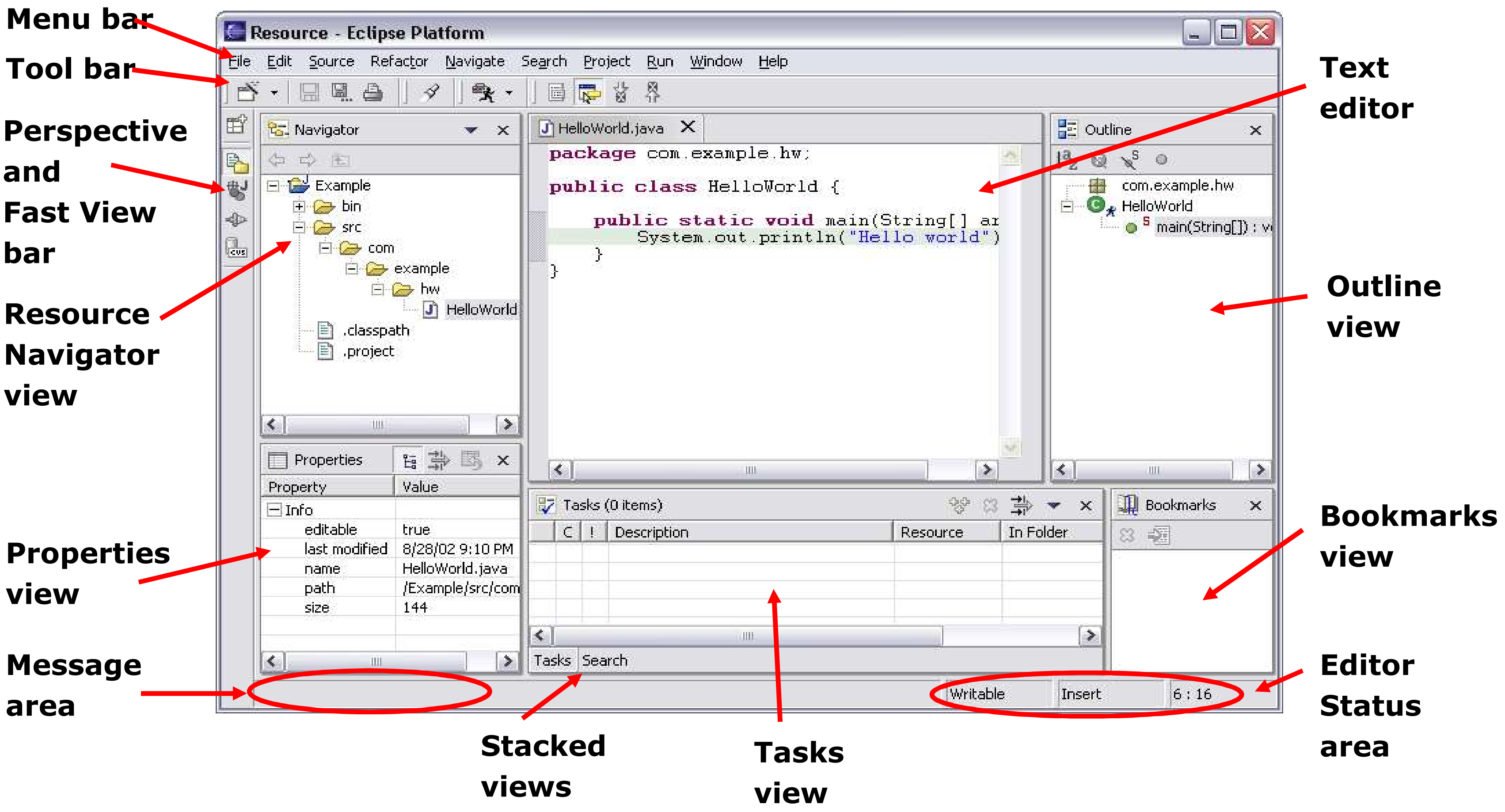
IDE is an application or set of tools that allows a programmer to write, compile, edit, and in some cases test and debug within an integrated, interactive environment

IDE combines:

- Editor
- Compiler
- Runtime environment
- debugger

© 2017 Capgemini. All rights reserved.

2.2 : Introduction to Eclipse IDE
Workbench Terminology



```
package com.example.hw;

public class HelloWorld {

    public static void main(String[] ar
        System.out.println("Hello world")
    }
}
```

Property	Value
Info	
editable	true
last modified	8/28/02 9:10 PM
name	HelloWorld.java
path	/Example/src/com
size	144

C	!	Description	Resource	In Folder

Writable | Insert | 6 : 16

Menu bar

Tool bar

Perspective and Fast View bar

Resource Navigator view

Properties view

Message area

Stacked views

Tasks view

Text editor

Outline view

Bookmarks view

Editor Status area

© 2017 Capgemini. All rights reserved.

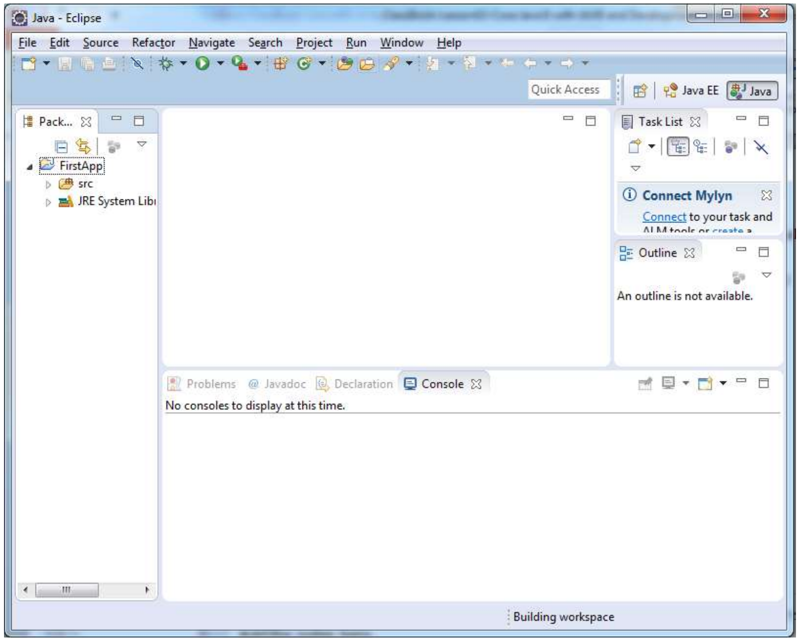
2.2 : Introduction to Eclipse IDE
The Workbench

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



Connect to your task and ALM tools or create a

An outline is not available.

No consoles to display at this time.

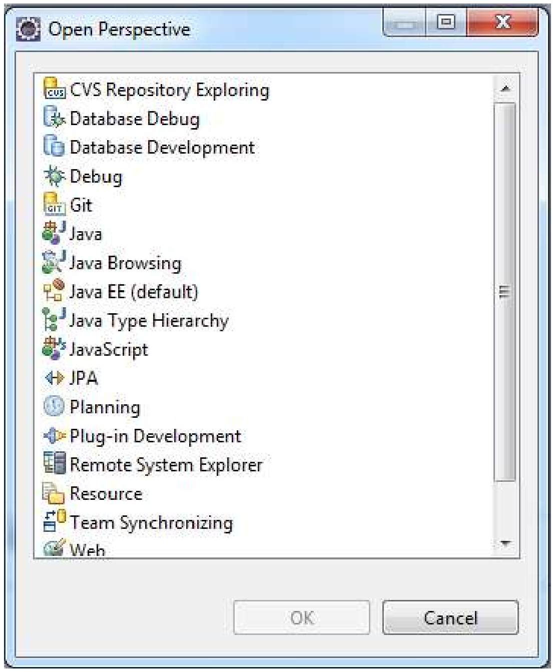
Building workspace

© 2017 Capgemini. All rights reserved.

2.2 : Introduction to Eclipse IDE
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



Open Perspective

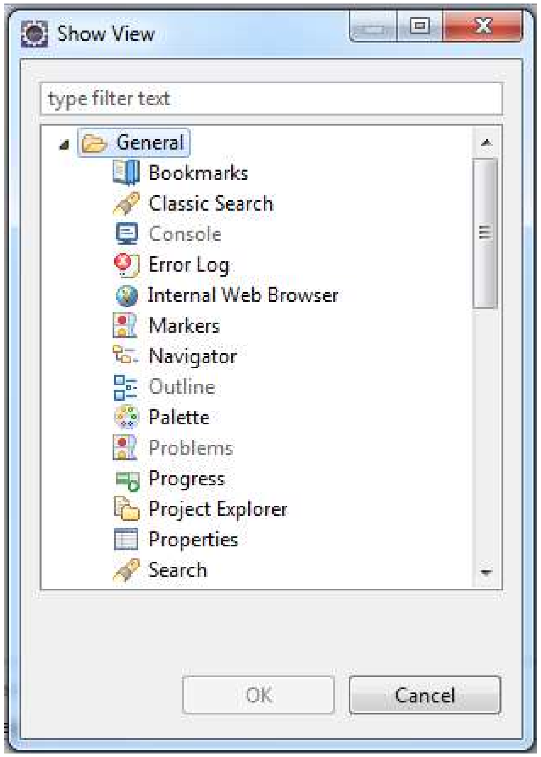
- CVS Repository Exploring
- Database Debug
- Database Development
- Debug
- Git
- Java
- Java Browsing
- Java EE (default)
- Java Type Hierarchy
- JavaScript
- JPA
- Planning
- Plug-in Development
- Remote System Explorer
- Resource
- Team Synchronizing
- Web

© 2017 Capgemini. All rights reserved.

2.2 : Introduction to Eclipse IDE
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

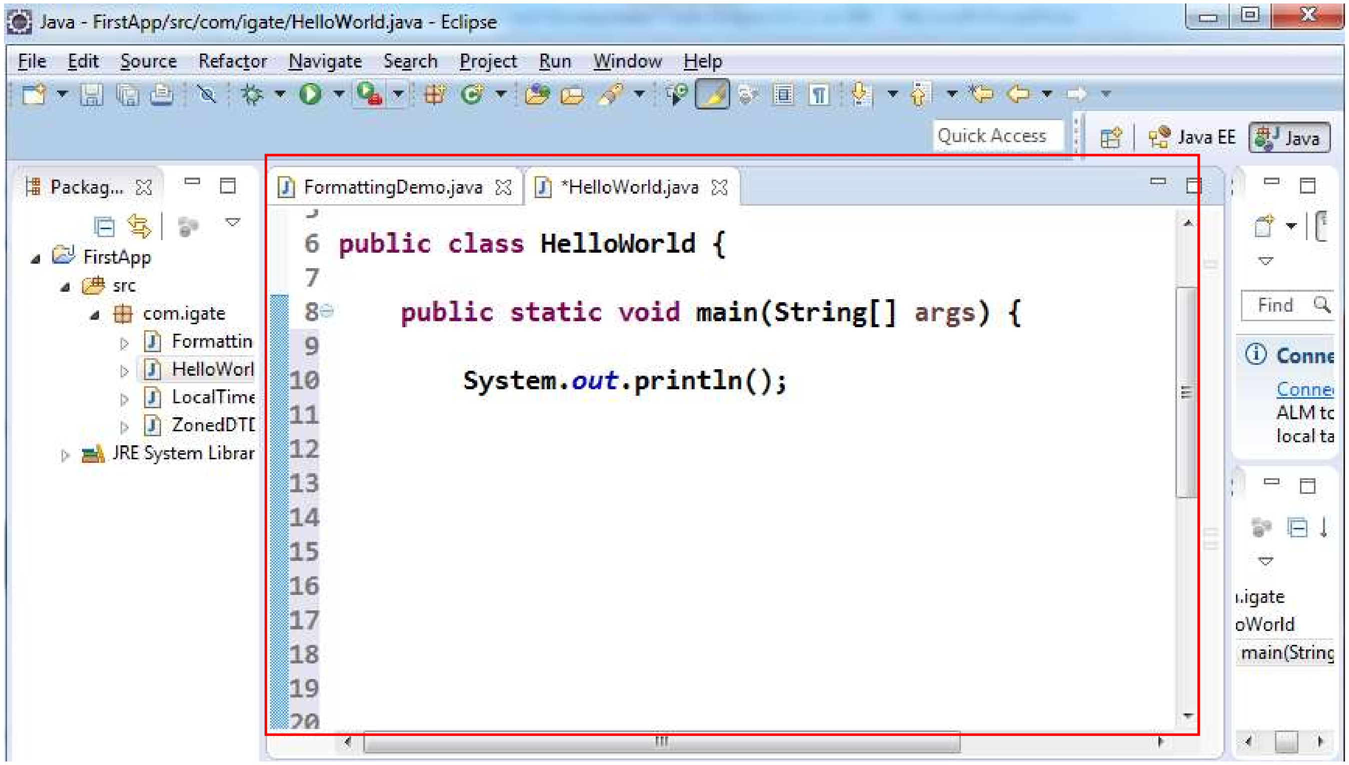


© 2017 Capgemini. All rights reserved.

2.2 : Introduction to Eclipse IDE
The Workbench

Editor:

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



```java
public class HelloWorld {

	public static void main(String[] args) {

		System.out.println();
```

© 2017 Capgemini. All rights reserved.

2.2 : Introduction to Eclipse IDE
The Workbench

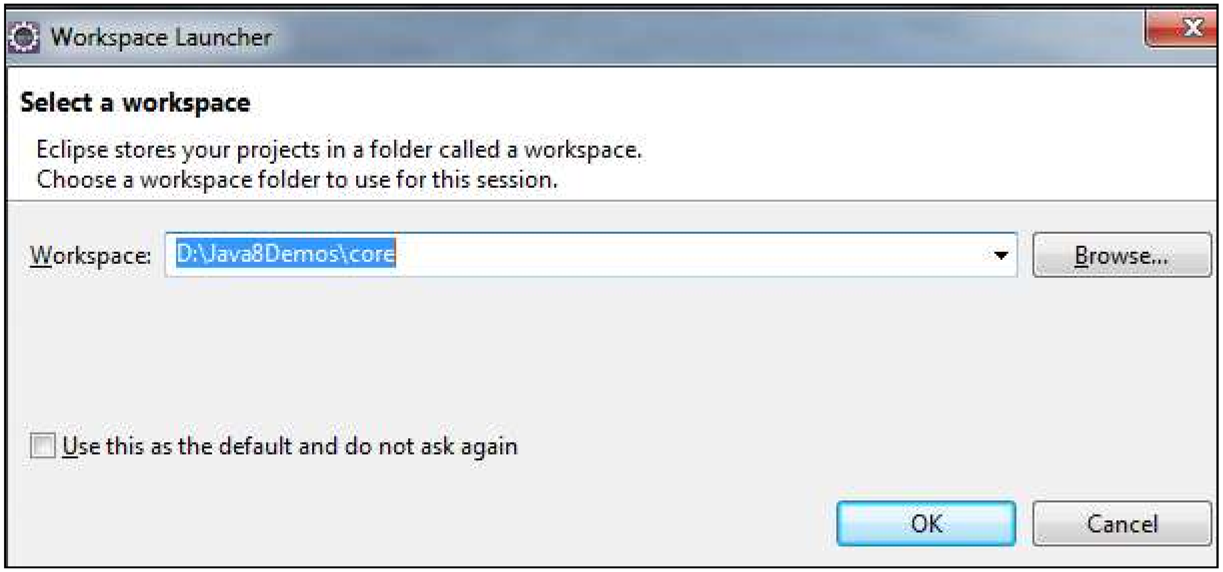
Notes page

© 2017 Capgemini. All rights reserved.

2.3: Creating and Managing Java Projects
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



Select a workspace

Eclipse stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

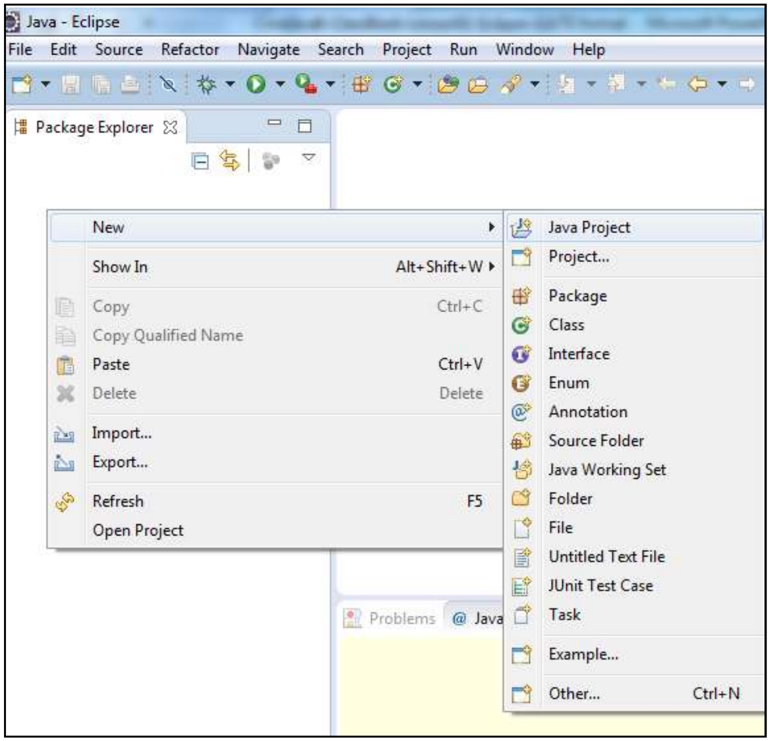
Workspace: D:\Java8Demos\core

□ Use this as the default and do not ask again

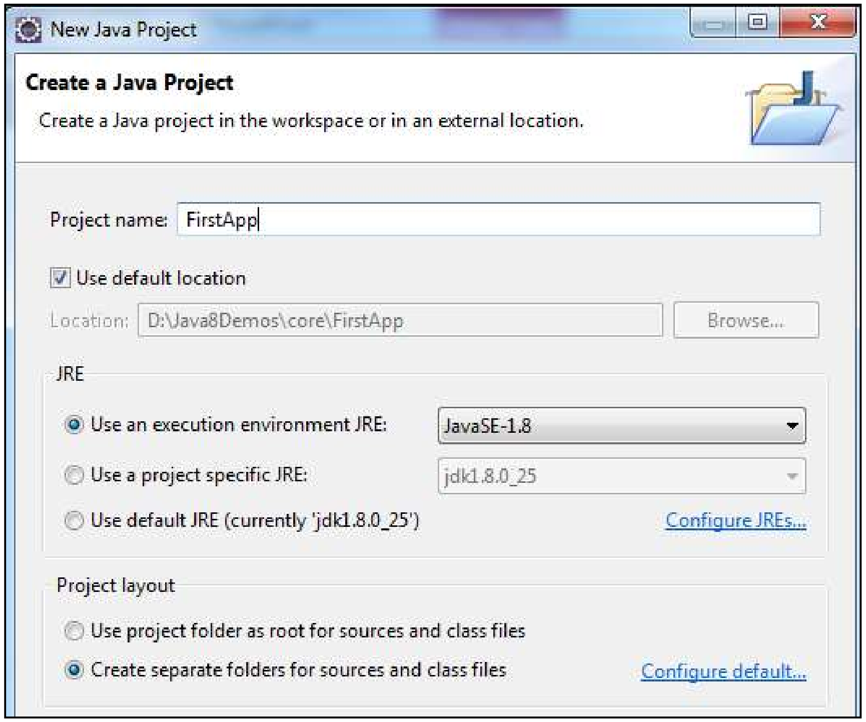
© 2017 Capgemini. All rights reserved.

2.3: Creating and Managing Java Projects
Create a Java Project

Right-click the Package Explorer panel, and select New-JavaProject.



Select Java project and provide a Project Name.



Create a Java Project

Create a Java project in the workspace or in an external location.

Project name: FirstApp

☑ Use default location

Location: D:\Java8Demos\core\FirstApp

JRE

● Use an execution environment JRE: JavaSE-1.8

○ Use a project specific JRE: jdk1.8.0_25

○ Use default JRE (currently 'jdk1.8.0_25')

Project layout

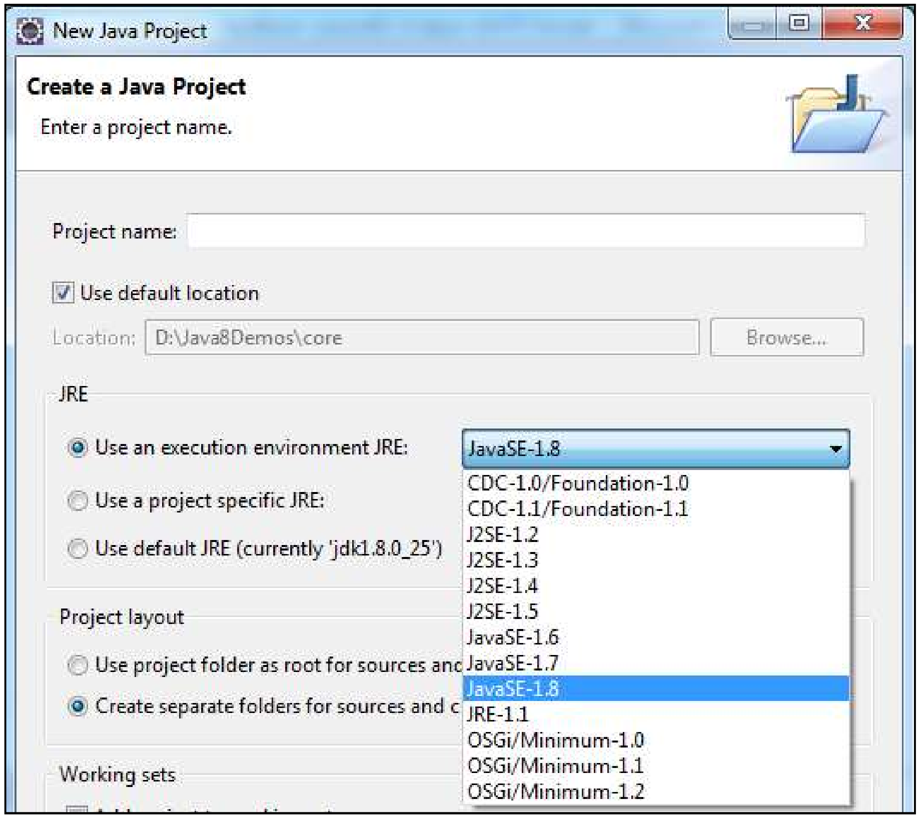
○ Use project folder as root for sources and class files

● Create separate folders for sources and class files

© 2017 Capgemini. All rights reserved.

2.3: Creating and Managing Java Projects
Select the JRE

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



New Java Project

Create a Java Project

Enter a project name.

Project name:

☑ Use default location

Location: D:\Java8Demos\core

JRE

● Use an execution environment JRE: JavaSE-1.8

○ Use a project specific JRE:

○ Use default JRE (currently 'jdk1.8.0_25')

Project layout

○ Use project folder as root for sources and

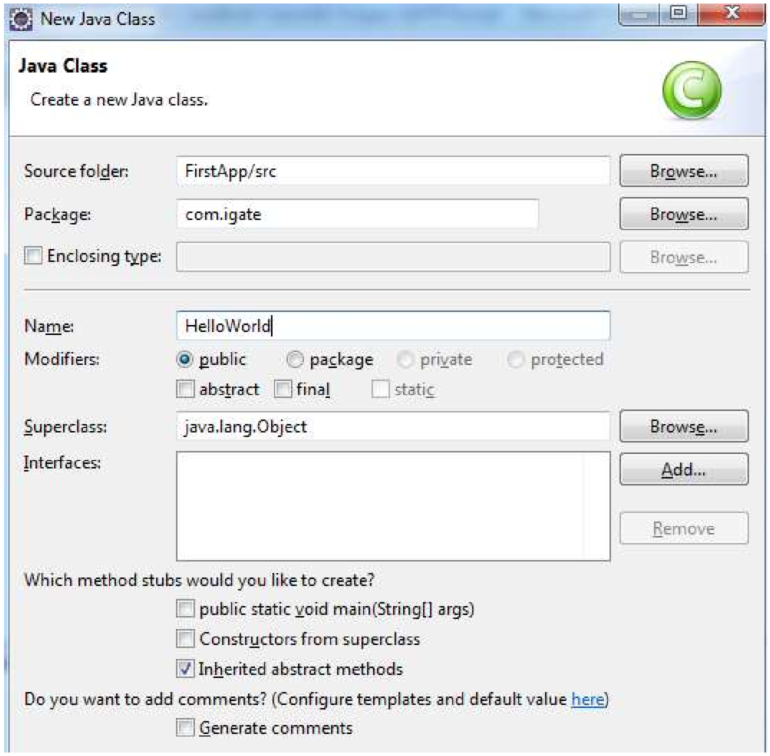
● Create separate folders for sources and c

Working sets

© 2017 Capgemini. All rights reserved.

2.3: Creating and Managing Java Projects
My first Java Program – Hello World

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



Java Class

Create a new Java class.

Source folder: FirstApp/src

Package: com.igate

□ Enclosing type:

Name: HelloWorld

Modifiers: ● public ○ package ○ private ○ protected

□ abstract □ final □ static

Superclass: java.lang.Object

Interfaces:

Which method stubs would you like to create?

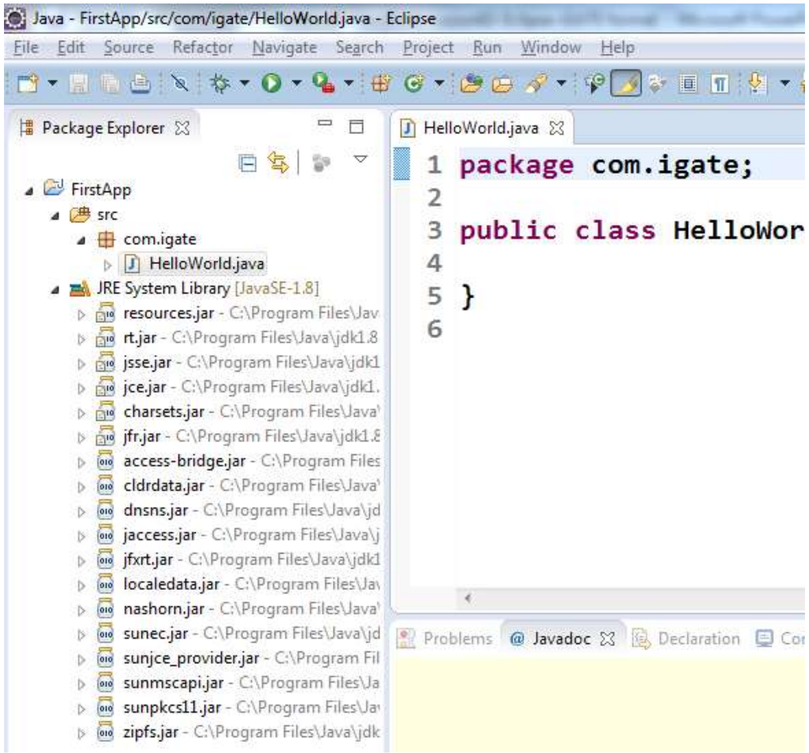
□ public static void main(String[] args)

□ Constructors from superclass

☑ Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)

□ Generate comments



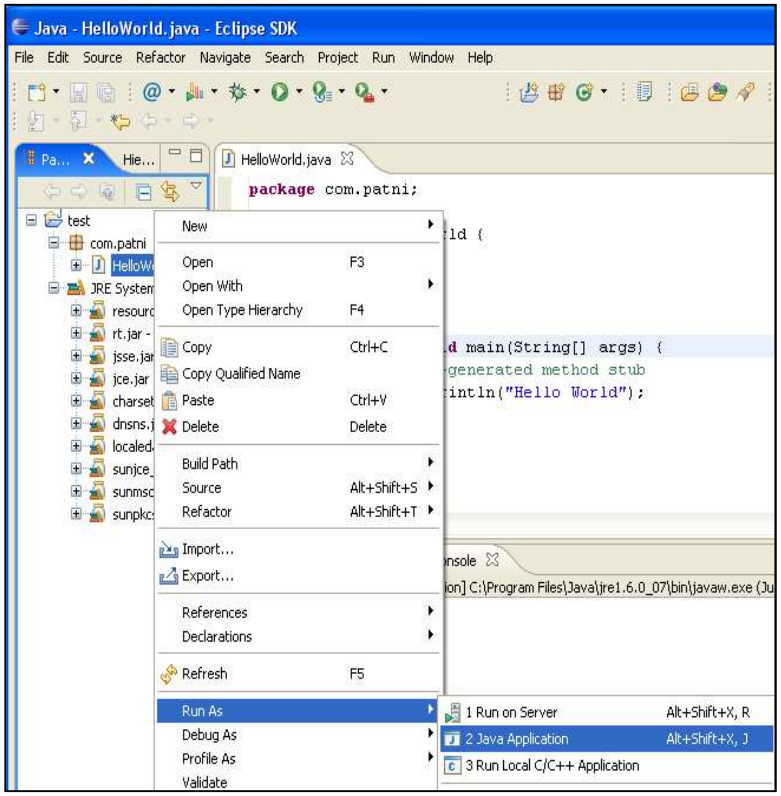
- FirstApp
  - src
    - com.igate
      - HelloWorld.java
  - JRE System Library [JavaSE-1.8]
    - resources.jar - C:\Program Files\Jav
    - rt.jar - C:\Program Files\Java\jdk1.8
    - jsse.jar - C:\Program Files\Java\jdk1
    - jce.jar - C:\Program Files\Java\jdk1.
    - charsets.jar - C:\Program Files\Java\
    - jfr.jar - C:\Program Files\Java\jdk1.8
    - access-bridge.jar - C:\Program Files
    - cldrdata.jar - C:\Program Files\Java\
    - dnsns.jar - C:\Program Files\Java\jd
    - jaccess.jar - C:\Program Files\Java\j
    - jfxrt.jar - C:\Program Files\Java\jdk1
    - localedata.jar - C:\Program Files\Jav
    - nashorn.jar - C:\Program Files\Java\
    - sunec.jar - C:\Program Files\Java\jd
    - sunjce_provider.jar - C:\Program Fil
    - sunmscapi.jar - C:\Program Files\Ja
    - sunpkcs11.jar - C:\Program Files\Jav
    - zipfs.jar - C:\Program Files\Java\jdk

```java
package com.igate;

public class HelloWor

}

```

© 2017 Capgemini. All rights reserved.

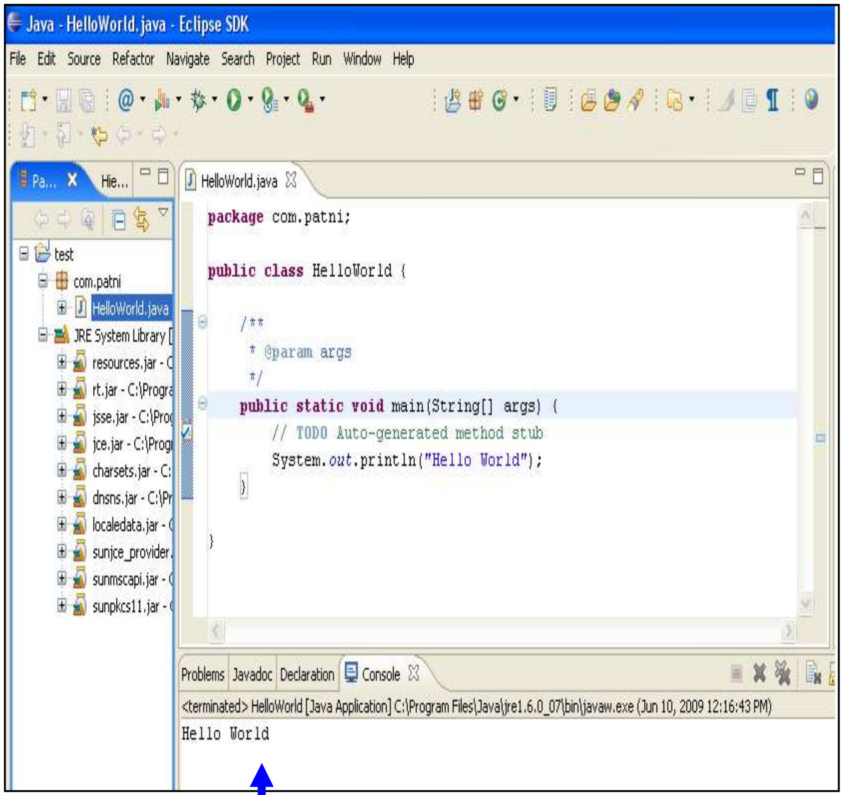
2.3: Creating and Managing Java Projects
Executing Hello World Program

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



```
package com.patni;

ld {

d main(String[] args) {
generated method stub
intln("Hello World");
```

```
ion] C:\Program Files\Java\jre1.6.0_07\bin\javaw.exe (Ju
```



```
package com.patni;

public class HelloWorld {

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        System.out.println("Hello World");
    }

}
```

```
<terminated> HelloWorld [Java Application] C:\Program Files\Java\jre1.6.0_07\bin\javaw.exe (Jun 10, 2009 12:16:43 PM)
Hello World
```

Output

© 2017 Capgemini. All rights reserved.

2.3: Creating and Managing Java Projects
Demo

HelloWorld Program using Eclipse IDE

© 2017 Capgemini. All rights reserved.

2.3: Creating and Managing Java Projects
Debugging your Java Program using Eclipse

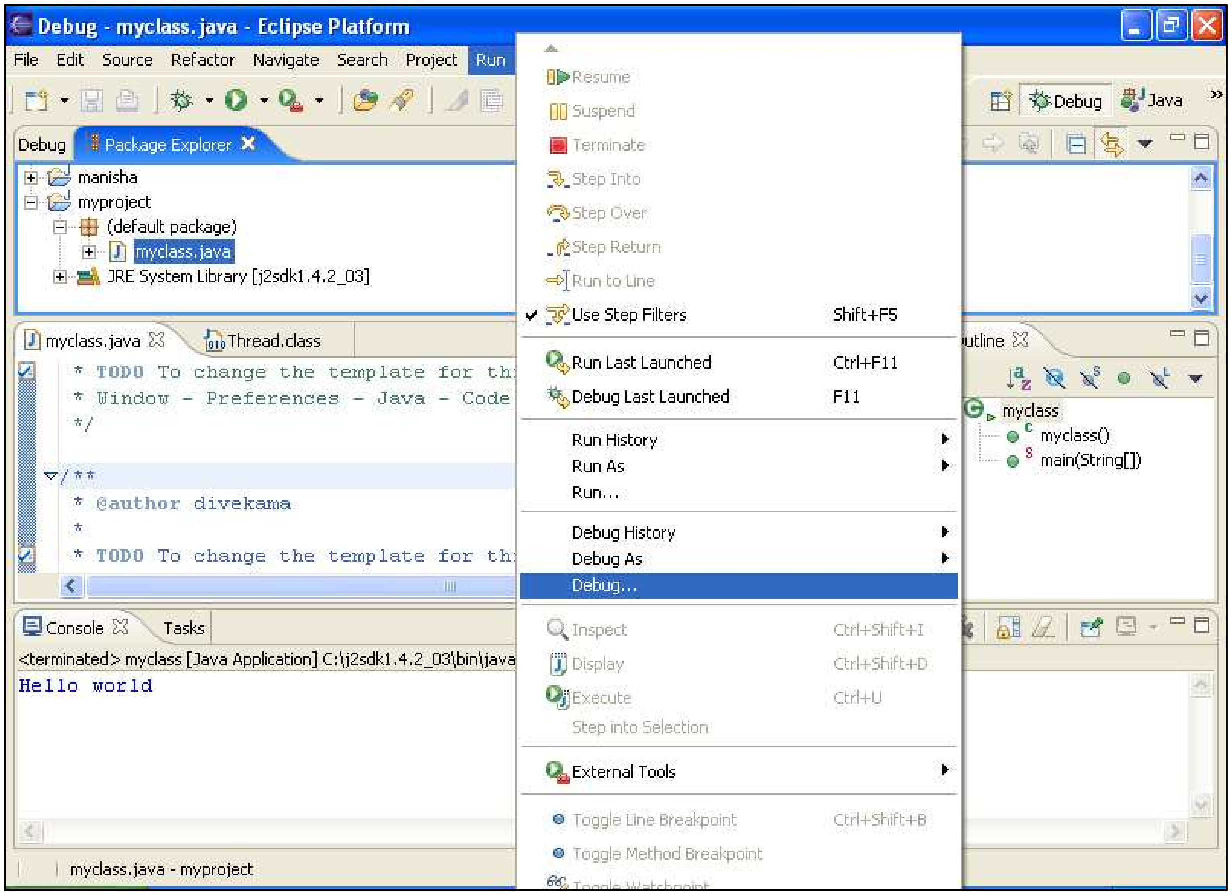
The Java Development Toolkit (JDT) includes a debugger that enables you to detect and diagnose errors in your programs running either locally or remotely

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

© 2017 Capgemini. All rights reserved.

2.3: Creating and Managing Java Projects
Debugging your Java Program using Eclipse

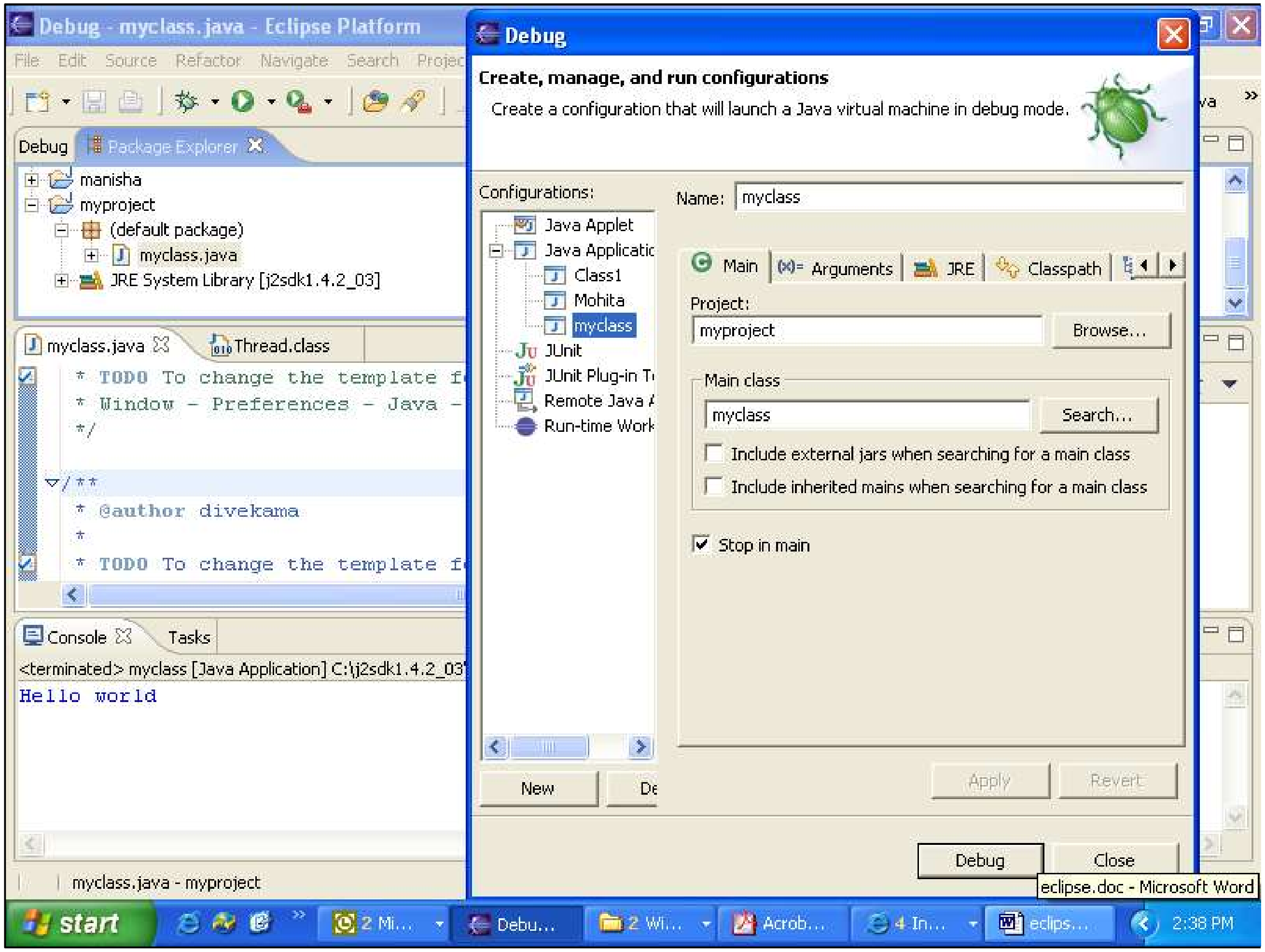


```
 * TODO To change the template for thi
 * Window - Preferences - Java - Code
 */

/**
 * @author divekama
 *
 * TODO To change the template for thi
```

```
<terminated> myclass [Java Application] C:\j2sdk1.4.2_03\bin\java
Hello world
```

© 2017 Capgemini. All rights reserved.

2.3: Creating and Managing Java Projects
Specifying Debugging options



```
 * TODO To change the template f
 * Window - Preferences - Java -
 */

/**
 * @author divekama
 *
 * TODO To change the template f
```

```
<terminated> myclass [Java Application] C:\j2sdk1.4.2_03
Hello world
```

Create, manage, and run configurations

Create a configuration that will launch a Java virtual machine in debug mode.

Name: myclass

Project: myproject

Main class: myclass

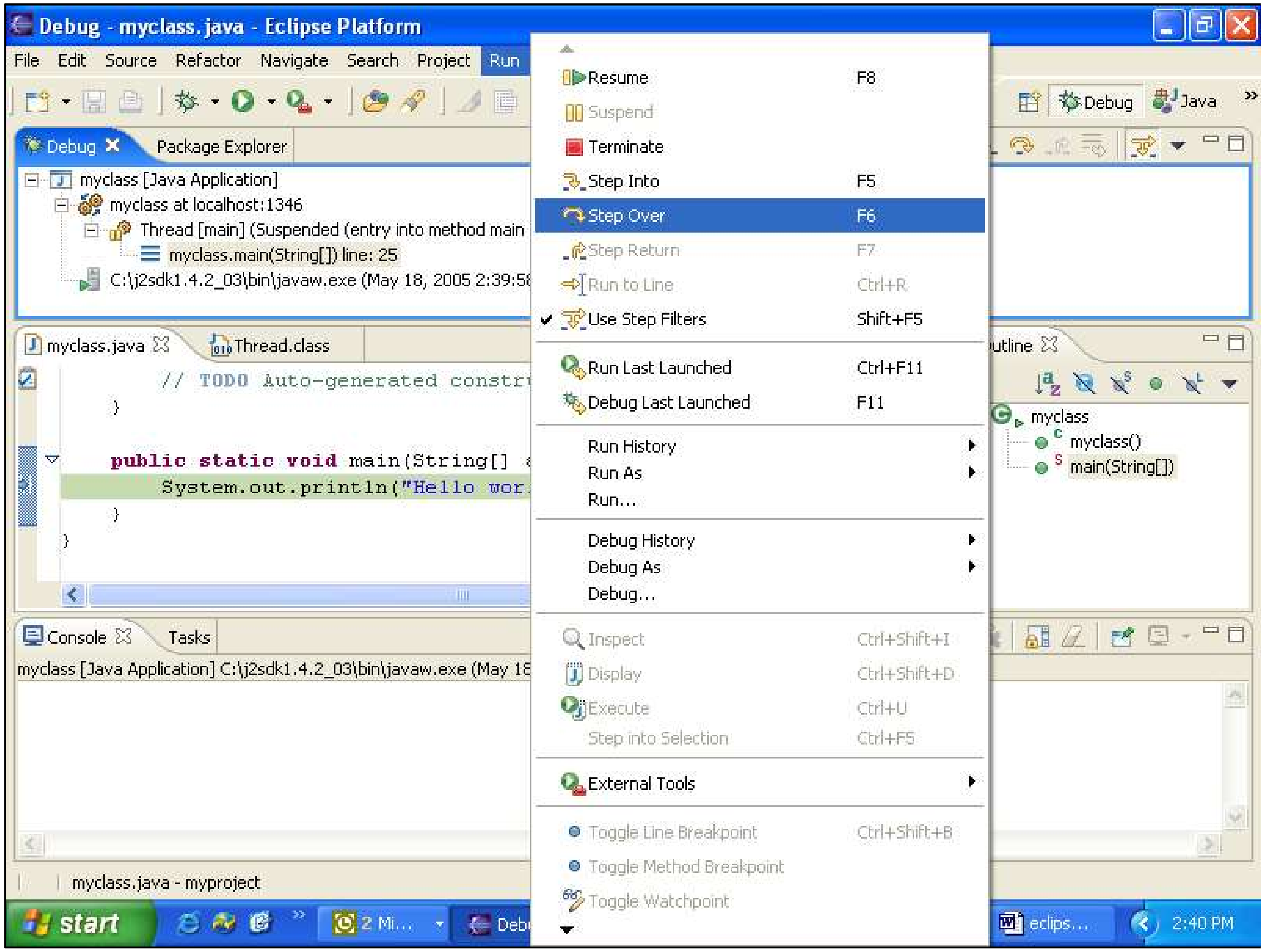
☐ Include external jars when searching for a main class

☐ Include inherited mains when searching for a main class

☑ Stop in main

© 2017 Capgemini. All rights reserved.

2.3: Creating and Managing Java Projects
Debugging a Java Program



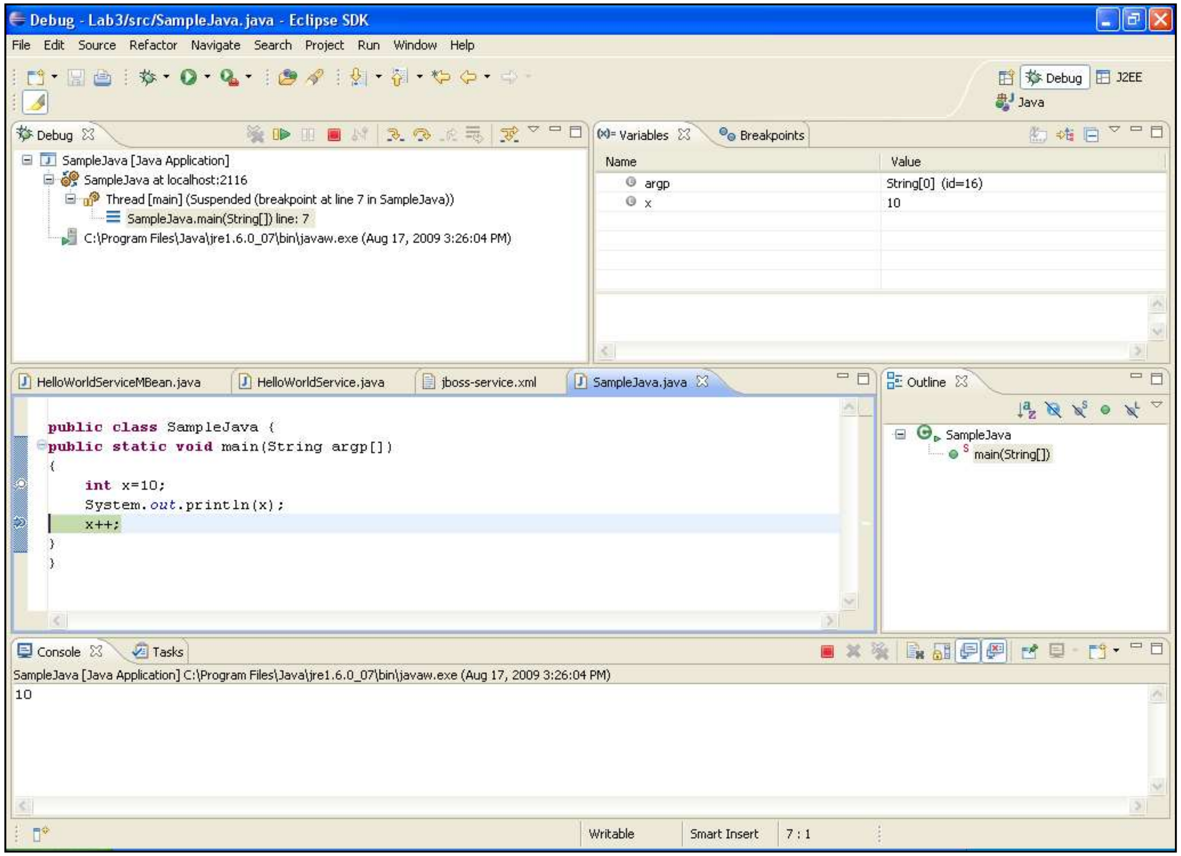
```
myclass [Java Application]
  myclass at localhost:1346
    Thread [main] (Suspended (entry into method main
      myclass.main(String[]) line: 25
  C:\j2sdk1.4.2_03\bin\javaw.exe (May 18, 2005 2:39:58
```

```
        // TODO Auto-generated constr
    }

    public static void main(String[] a
        System.out.println("Hello wor
    }
}
```

```
myclass [Java Application] C:\j2sdk1.4.2_03\bin\javaw.exe (May 18
```

© 2017 Capgemini. All rights reserved.

2.3: Creating and Managing Java Projects
Debugging a Java Program



```
SampleJava [Java Application]
  SampleJava at localhost:2116
    Thread [main] (Suspended (breakpoint at line 7 in SampleJava))
      SampleJava.main(String[]) line: 7
  C:\Program Files\Java\jre1.6.0_07\bin\javaw.exe (Aug 17, 2009 3:26:04 PM)
```

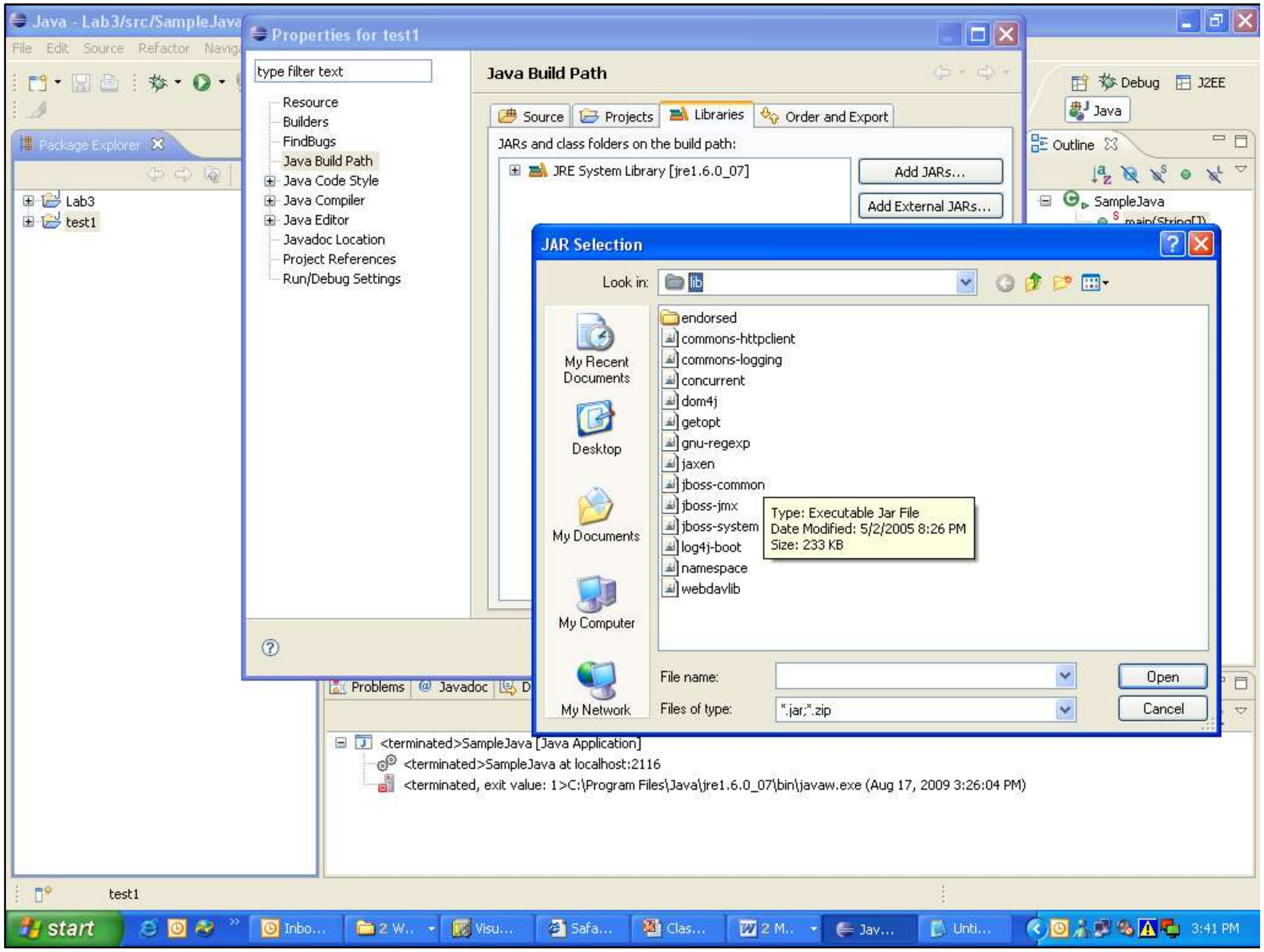
Name	Value
argp	String[0] (id=16)
x	10

```
public class SampleJava {
public static void main(String argp[])
{
    int x=10;
    System.out.println(x);
    x++;
}
}
```

```
SampleJava [Java Application] C:\Program Files\Java\jre1.6.0_07\bin\javaw.exe (Aug 17, 2009 3:26:04 PM)
10
```

© 2017 Capgemini. All rights reserved.

2.4: Miscellaneous Options
Adding external jar file



Type: Executable Jar File
Date Modified: 5/2/2005 8:26 PM
Size: 233 KB

```
<terminated>SampleJava [Java Application]
    <terminated>SampleJava at localhost:2116
    <terminated, exit value: 1>C:\Program Files\Java\jre1.6.0_07\bin\javaw.exe (Aug 17, 2009 3:26:04 PM)
```

© 2017 Capgemini. All rights reserved.

2.4: Miscellaneous Options
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

© 2017 Capgemini. All rights reserved.

2.4: Miscellaneous Options
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

© 2017 Capgemini. All rights reserved.

2.4: Miscellaneous Options
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:

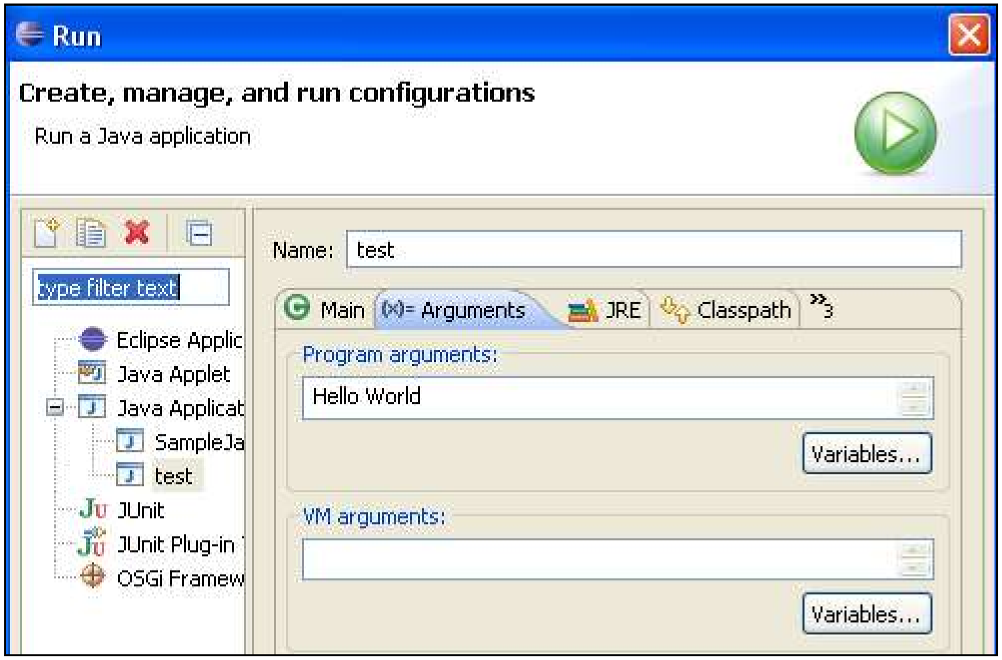
- **Window → Preferences → Java → Build Path → Classpath Variables**

© 2017 Capgemini. All rights reserved.

2.4: Miscellaneous Options
Passing Command Line Arguments

Command line arguments can be passed to the program in the following ways:

- Select **Run** → **Open Run** dialog → **Arguments** tab



Run

Create, manage, and run configurations

Run a Java application

Name: test

Program arguments:

Hello World

VM arguments:

© 2017 Capgemini. All rights reserved.

2.4: Miscellaneous Options
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

© 2017 Capgemini. All rights reserved.

2.4: Miscellaneous Options
Import a Project

Notes Page

© 2017 Capgemini. All rights reserved.

2.4: Miscellaneous Options
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

© 2017 Capgemini. All rights reserved.

2.4: Miscellaneous Options
Build options

Notes Page

© 2017 Capgemini. All rights reserved.

2.4: Miscellaneous Options
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

© 2017 Capgemini. All rights reserved.

2.4: Miscellaneous Options
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

© 2017 Capgemini. All rights reserved.

2.4: Miscellaneous Options
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

© 2017 Capgemini. All rights reserved.

2.4: Miscellaneous Options
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".

© 2017 Capgemini. All rights reserved.

Lab

Lab 1: Working with Java & Eclipse

© 2017 Capgemini. All rights reserved.

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

© 2017 Capgemini. All rights reserved.

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

© 2017 Capgemini. All rights reserved.