

Chulalongkorn University International School of Engineering Department of Computer Engineering 2140105 Computer Programming Lab. Name \_\_\_\_\_ Student ID \_\_\_\_\_ Station No. \_\_\_\_\_ Date \_\_\_\_\_

Lab 1 - An Introduction to Eclipse

## **Objectives:**

- Learn to use Eclipse as Java integrated development environment.
- Practice basic java programming.

# Background Knowledge

• Materials from 2140101 - Computer Programming for International Engineer.

# Eclipse - an open development platform

Eclipse is an open source community, whose projects are focused on building an open development platform comprised of extensible frameworks, tools and runtimes for building, deploying and managing software across the lifecycle. A large and vibrant ecosystem of major technology vendors, innovative start-ups, universities, research institutions and individuals extend, complement and support the Eclipse platform

In this class, we will use Eclipse as a tool (IDE) for developing JAVA programs.

You will need a Java runtime environment (JRE) to use Eclipse. You may download the Eclipse SDK at *http://www.eclipse.org/downloads/*. You will get a Zip file, uncompress (unzip) it in an empty folder. You are ready to run the Eclipse.

# The Workbench

When the Workbench is launched the first thing you see is a dialog that allows you to select where the workspace should be located as shown in Figure 1. The workspace is the directory where your work will be stored.



# Your turn 🔘

Since this is the first time you use Eclipse, you need to create a workspace. Your workspace will have the following name format, %desktop%\YourID, where %desktop% is the location of your desktop and YourID is your student identification. For example, if the desktop of your machine is "C:\Documents and Settings\Chate\Desktop", and your student ID is "500 12343 21", your work space will be

"C:\Documents and Settings\Chate\Desktop\5001234321".

You should create a folder for your work and copy it. We will clean the hard disk everyday. Do not expect your work will be in the computer next time you work in the lab.

You will use the workspace you've just created for all labs in this course.

After the workspace location is chosen, a single Workbench window is displayed as shown in Figure 2. A Workbench window offers one or more perspectives. A perspective contains editors and views, such as the Navigator. Multiple Workbench windows can be opened simultaneously. Initially, in the first Workbench window that is opened, the Java perspective is displayed, with only the Welcome view visible. Click the arrow labeled Workbench in the Welcome view to cause the other views in the perspective to become visible.

(You can get the Welcome view back at any time by selecting **Help > Welcome**.)



Figure 1 - Eclipse Workspace Launcher



Figure 2 - Eclipse Welcome page

# **Preparing Eclipse**

In this section, you will verify that Eclipse is properly set up for Java development.

## Verifying JRE installation and CLASSPATH variables

1. If you still see the Eclipse Welcome page, click the arrow icon to begin using Eclipse.

- Select the menu item <u>Window > Preferences...</u> to open the workbench preferences.
- 3. Select the Java > Installed JREs preference page to display the installed Java Runtime Environments as shown in Figure 3. Confirm that a JRE has been detected. By default, the JRE used to run the workbench will be used to build and run Java programs. It should appear with a checkmark in the list of installed JREs. We recommend that you use a Java SDK instead of a JRE. An SDK is designed for development and contains the source code for the Java library, easing debugging. Additional SDKs can be added by searching the hard drive for installed SDKs. To do so, simply click the Search... button and specify a root folder for the search. Select the most recently version of the JRE.

Installed JREs	⇔ • ⇒ •		
Add, remove or edit : By default, the check Installed <u>J</u> REs:	JRE definitions. ed JRE is added to the build path	of newly created	Java projects.
Name	Location	Туре	<u>A</u> dd
🗹 🛋 Java142	C:\Program Files\IBM\Java142	Standard VM	<u>E</u> dit
			<u>⊆</u> ору
			Remove
			Search

Figure 3 - Installed JREs preference page

4. Select the <a>General > Workspace</a> preference page as shown in Figure 4. Confirm that the **Build automatically** option is checked.



Figure 4 - Workspace preference page

5. Select the <a>Java > Build Path</a> preference page as shown in Figure 5. Confirm that Source and output folder is set to Project.

Build Path	<b>() →</b> → →
Specify the build path entri wizard:	es used as default by the New Java Project creation
Source and output folder O Project C Folders	
Source folder name:	src
Output folder name:	bin
As JRE library use: JRE co	ntainer

Figure 5 - Java's Build Path preference page

6. Select the <a>[w]</a>Java > Editor preference page as shown in Figure 6. Confirm that option Report problems as you type is checked.

Editor	⇔ • ⇔ •
Java editor preferences. Note that some preferences may be sel <u>Editors</u> preference page.	t on the <u>Text</u>
Smart caret positioning at line start and end	
🗹 Smart caret positioning in Java names (overrides platform be	ehavior)
Report problems as you type	
Also see the <u>compiler warnings</u> and <u>spell checking</u> preference	es.

Figure 6 - Java's Editor preference page

7. Click on **OK** to save the preferences.

**Creating your first Java project** 



## Your turn 🛈

Complete the following steps.

1. Creating a project.

Before we start working on a program, we must create a project for it. An Eclipse project helps us organize our files, so that we do not mistakenly work on the wrong program.

- 1.1. To create a project, go to **File** and choose **new -> project** as shown in Figure 7.
- 1.2. Since we are writing a Java program, choose Java Project.
- 1.3. Then type in your project names. Let's name it **Lab1v1**. The name that we use is case sensitive.

1.4. At this point, we have two choices, **next** and **finish**. This is a simple project, so let's choose **finish**. Our project will then be displayed on the left hand window. The window is divided into 4 parts as shown in Figure 8. We can extend or move all parts to anywhere we want.

File Edit	Source	Refactor	Navigate	Search	Project	Run	Window	Help
New		Alt+S	hift+N 🔹 🕨	📑 Proje	ect			þ 🗘 🔹
Close Close	All	Ctrl+F Ctrl+S	=4 5hift+F4	🔡 Pack	age s			
🔚 Save 📓 Save	As	Ctrl+S	5	🕜 Inte	rface rce Folder			
Reve	All rt	Ctrl+9	5hift+S	C Fold	er			
Move Rena Refre	 me sh	F2 F5		📑 JUni 🤏 Visua	t Test Ca: al Class	se		
🖹 Print.		Ctrl+F	5	Exar	nple			
🖆 Print. Swite	n Workspa	Ctrl+f ce	2	📑 Othe	er		Ctrl+N	

Figure 7 - Create a new project



**Tip**: Right clicking in the Eclipse Explorer window will also get you a pop-up menu for creating your project.

- 9. Creating your file.
  - 9.1. Try creating you class (yes, it's your program) by choosing **File-> New ->Class**. You can also do this by selecting from pop-up menu if you right click in the explorer window.
  - 9.2. A new Java class dialog will pop up, prompting you to fill in your class information as shown in Figure 9.
  - 9.3. Let's create a class called "*Hello*". Let this class have its main method. <u>Do not</u> select Inherit abstract method. You can simply select the check boxes. Eclipse will generate your main method structure for you, with lots of comments. You can leave those comments in.
  - 9.4. Let's write a program to print the word "Hello!" This is easy, but you've got to get used to Eclipse, *so do it now.*
  - 9.5. While you are writing your program, you may notice that your file name has a \* in it. This means your file has not been saved.

Java Class	efault package is discouraged.	C
Source folder:	Lab1/src	Browse
Package:	(default)	Browse
Enclosing type:	util.Keyboard	Browse
Name:	Hello	
Modifiers:	public O default O private O protected     abstract I final I static	
Superclass:	java.lang.Object	Browse
Interfaces:		Add
	[	Remove
Which method stut	os would you like to create? public static void main(String[] args) Constructors from superclass herited abstract methods	
Do you want to add	Generate comments	ojecti

Figure 9 - New Java Class dialog

9.6. When you finish writing your program, try to run it without saving (select Run -> Run As Java Application). Eclipse will prompt you to save your file at this point. What you print will be shown in Eclipse's console window.

**NOTE**: Sometimes if your program won't run, try to save it manually first. This may fix the problem.

#### 10. Importing file.

Sometimes you won't be writing your program from scratch. Importing is a way to copy files into your project. You can do this by right click on your project name, and then choose import. Eclipse will pop a form prompting you to choose what kind of information will be imported as shown in Figure 10. We will mainly be using 2 import sources:

- file system normal copying of files into your project.
- archive file import an unzip the *.zip* or *.jar* file (jar is Java's zip format) for you.

🗢 Import	
Select Choose import source.	Ľ
Select an import source:	
type filter text	
Image: Constraint of the second se	
O < Back Next >	Finish Cancel

Figure 10 - Import dialog

- 10.1. Try importing the InsertionSort.jar file. (You may ask Lab. Instructor for a copy of InsertionSort.jar)
- 10.2. Now, unzip the jar to any other folder, create a new project called *Lab1v2* and try to import the unzipped file.
- 11. Creating .jar file

Actually, jar (java archive) is zip, but we can also make .jar file executable. This will be useful when you want to deploy your program in real use.

- 11.1. To create a jar file, right click on your project, and then select Export as shown in Figure 11. Choose JAR file, name it (click Browse and change the folder to Desktop, and type in yourname.jar), and select all options as shown in Figure 12 then click Next.
- 11.2. JAR Package option dialog is shown, click Next.
- 11.3. JAR Menifest Specification dialog is shown as in Figure 13. Click Browse.

- 11.4. Select main class window is shown. Select *InsertionSort* and then click **OK** as shown in Figure 14.
- 11.5. Click Finish.
- 11.6. Go to Windows desktop and verify that your JAR file created in step 11.1 exists. Double-click that file to run the executable jar file.

🛢 Export			
Select Package resources into a	JAR file on the loc	al file system	凸
Select an export destinat	ion:		
* Ant Buildfiles	and fragments		

Figure 11 - Export dialog



Figure 12 - Export JAR file

JAR Export		*
JAR Manifest Specification	-	
Customize the manifest file for the JAR file.	-	
Specify the manifest:		
Generate the manifest file		
Save the manifest in the workspace		
Beuse and save the manifest in the workspace		
Manifest file:	Brgwse	3
Mandest file:	Browse-	]
Seal contents:	Details	-
Seal some gackages     N	lothing sealed Dgtails	j
Select the class of the application entry point:		
Main glass:	Browge-	$\mathbb{D}$
		-
⑦ < <u>Back</u> Next >	Einish Cancel	٦

Figure 13 - JAR Manifest Specification

Select the class which is the app	lication's entry point:
⊕ InsertionSort     ⊕ SelectionSortDemo	
(default package) , tert	
(default package) - test	
(default package) - test	

Figure 14 - Select main class window

#### NOTE:

- When you submit your homework, use .jar format, you must choose to export generated class files and resources and export Java source files and resources.
- Before sending your jar file to anywhere, try unzipping it to test if the jar creation is correct.

### **Tips & Tricks**

- 1. To let the editor show line numbers:
  - 1.1. Go to menu Window, choose Preferences.
  - 1.2. At **General -> Editors -> Text Editors**, check show line numbers as shown in Figure 15.

General       Indo history size:       200         Appearance       Capabilities       Displayed tab width:       4         Compare/Patch       Insert spaces for tabs       Displayed tab width:       4         Content Types       Insert spaces for tabs       Insert spaces for tabs       Insert spaces for tabs         Editors       File Associations       Insert spaces for tabs       Insert spaces for tabs         Structured Text Ec       File Associations       Print margin         Network Connections:       Perspectives       Search         Perspectives       Image: Indicator       Show whitespace characters         Startup and Shutdown       Velorme       Velorme         Welcorne       Vorkspace       Ant         Connectivity       Ecore Diagram       Find scope         Internet       Selection foreground color       Color:         Java       JPA       Find scope         JPA       Background color       Selection background color         Plug-in Development       More colors can be configured on the Colors and Fonts preference page         Server       COA Lools       More colors can be configured on the Colors and Fonts preference page	e filter text	Text Editors			<b>\$</b> • =
Capabilities       Displayed tab width:       4         Compare/Patch       Insert spaces for tabs         Content Types       Imsert spaces for tabs         Editor       File Associations         Structured Text Ec       Print margin         Text Editors       Print margin         Keys       Show print margin         Network Connections       Print margin column         Perspectives       Image indicator         Startup and Shutdow       Image indicator         Web Browser       Image indicator         Welcome       Show affordance in hover on how to make it sticky         Workspace       Ant         Connectivity       Enselle of reground color         Install/Update       Find scope         Internet       Selection background color         Java       Java         JPA       More colors can be configured on the <u>Colors and Fonts</u> preference page         Server       Scolar can be configured on the <u>Colors and Fonts</u> preference page	General	Undo history size:	200	3	
Compare/Patch Content Types       Insert spaces for tabs         Editor       File Associations         Structured Text Ec Text Editors       Print margin columnt 80         Keys       Show print margin columnt 80         Network Connections       Print margin columnt 80         Perspectives       Image indicator         Startup and Shutdown       Image indicator         Web Browser       Image indicator         Welcome       Show affordance in hover on how to make it sticky         Workspace       Ant         Ant       Appearance color options:         Connectivity       Current line highlight Print margin         Install/Update       Find scope         Internet       Selection background color         Java       Background color         JPA       Foreground color         Model Validation       Hypelink         Plug-in Development       More colors can be configured on the <u>Colors and Fonts</u> preference page         Server       Scola Loot	Capabilities	Displayed tab width:	4		
Content Types       Image: Highlight current line         Editors       Show print margin         Structured Text Ec       Point margin column:         Keys       Show print margin         Network Connections       Point margin column:         Perspectives       Show whitespace characters         Search       Image: Indicator         Startup and Shutdowi       Image: Indicator         Web Browser       Image: Indicator         Welcome       Show affordance in hover on how to make it sticky         Workspace       Ant         Connectivity       Ecore Diagram         Help       Find scope         Internet       Selection background color         Java       Java         JPA       Foreground color         Model Validation       Hypelink         Plug-in Development       More colors can be configured on the <u>Colors and Fonts</u> preference page         Server       Scota case         Server       Scota case	Compare/Patch	Insert spaces for tabs			
Editors         File Associations         Structured Text Ec         Text Editors         Keys         Network Connection:         Perspectives         Search         Startup and Shutdown         Web Browser         Welcome         Workspace         Ant         Connectivity         Ecore Diagram         Help         Internet         Java         JPA         Model Validation         PluA         Model Validation         Plus         More colors can be configured on the <u>Colors and Fonts</u> preference page         Server         SCOL Teacl	Content Types	Highlight current line			
File Associations         Structured Text Ec         Text Editors         Keys         Network Connections         Perspectives         Search         Startup and Shutdown         Web Browser         Welcome         Workspace         Ant         Connectivity         Ecore Diagram         Help         Install/Update         Internet         Java         JPA         Model Validation         Plus         Model Validation         Plus         More colors can be configured on the <u>Colors and Fonts</u> preference page         Server         SCAL Teact	Editors	Show print margin			
Text Editors         Keys         Network Connection:         Perspectives         Search         Startup and Shutdown         Web Browser         Welcome         Workspace         Ant         Connectivity         Ecore Diagram         Help         Install/Update         Internet         Java         JPA         Model Validation         Plua, in Development         Run/Debug         Server         Scota         Scota         Server         Scota         Scota	Structured Text Ec	Print margin column:	80	1	
Keys       Show range indicator         Perspectives       Show whitespace characters         Search       Show whitespace characters         Startup and Shutdow       We fore editing a derived file         Welcome       Show affordance in hover on how to make it sticky         Workspace       Ant         Connectivity       Show affordance in hover on how to make it sticky         Keys       Color:         Internet       Selection foreground color         Java       Selection background color         JPA       Foreground color         Model Validation       Hypefink         Plug-in Development       More colors can be configured on the <u>Colors and Fonts</u> preference page         Server       Scota cale	Text Editors	Show line numbers	1.55		
Network Connection:       Show whitespace characters         Perspectives       Show whitespace characters         Search       Image: module of text         Startup and Shutdow       Image: module of text         Web Browser       Image: module of text         Welcome       Image: module of text         Welcome       Image: module of text         Welcome       Image: module of text         Workspace       Ant         Connectivity       Enable drag and drop of text         Ecore Diagram       Appearance color options:         Help       Find scope         Internet       Selection foreground color         Java       Selection background color         Java       Selection background color         Java       Background color         Java       More colors can be configured on the <u>Colors and Fonts</u> preference page         Server       Scolar         Server       Scolar	Keys	Show range indicator			
Perspectives Search       Image: Search Startup and Shutdown Web Browser         Web Browser       Image: Smart caret positioning at line start and end         Welcome       Image: Smart caret positioning at line start and end         Welcome       Image: Smart caret positioning at line start and end         Welcome       Image: Smart caret positioning at line start and end         Welcome       Image: Smart caret positioning at line start and end         Workspace       Ant         Connectivity       Eine number foreground         Ecore Diagram       Current line highlight         Help       Print margin         Internet       Selection foreground color         Java       Selection foreground color         Java       Background color         Java       Background color         Java       More colors can be configured on the <u>Colors and Fonts</u> preference page         Server       Scotars         Scotars       Teach	Network Connections	Show whitesnace char	acters		
Search       Image: Construction of the second	Perspectives	Enable drag and drog	of text		
Startup and shurbowi       With before country a defined me.         Web Browser       Image: Smart caret positioning at line start and end         Web Come       Image: Smart caret positioning at line start and end         Web Show affordance in hover on how to make it sticky         Workspace       Ant         Connectivity       Ecore Diagram         Help       Current line highlight         Print margin       Find scope         Internet       Selection foreground color         Java       Background color         JPA       Foreground color         More colors can be configured on the <u>Colors and Fonts</u> preference page         Server       Scolar Scope         Scolar Scope       Scolar Scope	Search	Warn before edition a	darived file		
Web brokser       Web brokser         Welcome       Welcome         Workspace       Ant         Ant       Appearance color options:         Connectivity       Current line highlight         Ecore Diagram       Current line highlight         Help       Print margin         Install/Update       Selection foreground color         Java       Background color         JPA       Foreground color         Model Validation       Hypeflink         Plug-in Development       More colors can be configured on the <u>Colors and Fonts</u> preference page         Server       Scon to be configured on the <u>Colors and Fonts</u> preference page	Web Browser	Smart caret nositionin	a at line start and en	d	
Workspace         Ant         Connectivity         Ecore Diagram         Help         Install/Update         Java         JPA         Poleston         More colors can be configured on the Colors and Fonts preference page         Selection         Plug-in Development         Run/Debug         Server         Scot         Scot         Server         Scot         Scot         Server         Scot         Scot         Scot         Server         Scot	Welcome	Show affordance in ho	yer on how to make	uit sticky	
Ant Appearance color options: Connectivity Ecore Diagram Help Install/Update Internet Java JPA Nodel Validation Plug-in Development Run/Debug Server	Workspace	C Show and Gallet in the		in society	
Connectivity       Line number foreground       Color:         Ecore Diagram       Current line highlight       Print margin         Help       Print margin       Find scope         Install/Update       Selection foreground color       Selection background color         Java       Background color       Selection background color         JPA       Foreground color       Hyperlink         Plug-in Development       More colors can be configured on the <u>Colors and Fonts</u> preference page         Server       Sectore Defaults       App	Ant	Appearance color options	E.		
Ecore Diagram Help Print margin Find scope Selection foreground color Java Java JPA Foreground color JPA Foreground color Hyperlink Plug-in Development Run/Debug More colors can be configured on the <u>Colors and Fonts</u> preference page Server Server Secon Fonts Restore Defaults Apple	Connectivity	Line number foreground		Color:	
Heip     Print margin       Install/Update     Find scope       Internet     Selection foreground color       Java     Background color       JPA     Foreground color       Model Validation     Hyperlink       Plug-in Development     More colors can be configured on the <u>Colors and Fonts</u> preference page       Server     Sectore Defaults	Ecore Diagram	Current line highlight			
Install Opdate Internet Selection foreground color Selection background color Java Java JPA Foreground color Hyperlink Plug-in Development Run/Debug More colors can be configured on the <u>Colors and Fonts</u> preference page Server Server Secon Fonts Restore Defaults Appl	Help Install/I Indate	Find scope			
Java Java Java Selection background color Background color Background color Foreground color Hyperlink Plug-in Development Run/Debug More colors can be configured on the <u>Colors and Fonts</u> preference page Server Server Background color Colors and Fonts preference page Server Background color Colors and Fonts page Server B	Install/Opdate	Selection foreground col	lor		
JPA Foreground color Model Validation Hyperlink Plug-in Development Run/Debug More colors can be configured on the <u>Colors and Fonts</u> preference page Server Server	Java	Selection background co	lor		
Model Validation         Hyperlink           Plug-in Development         More colors can be configured on the <u>Colors and Fonts</u> preference page           Server         Server	JPA	Foreground color			
Plug-in Development Run/Debug More colors can be configured on the <u>Colors and Fonts</u> preference page Server COA Table Restore Defaults Ann	Model Validation	Hyperlink			
Run/Debug More colors can be configured on the <u>Colors and Fonts</u> preference page Server - Restore Defaults Ann	Plug-in Development				
Server Rectore Defaults Ann	Run/Debug	More colors can be config	gured on the Colors	and Fonts prefere	nce page.
Restore Lietauts App	Server .		6	unterer Defaulte	Analy
III )	m. ) •		[n	ESTORE DEMUIS	Abbiy

Figure 15 - Text Editors preference page

- 2. If you want to find a bracket's corresponding pair, move the cursor behind the bracket. Its pair will be shown.
- 3. Use **Ctrl+Shift+F** to layout your code beautifully.
- 4. If you are typing a long name, while typing, press **Ctrl+Space**. Eclipse will fill the rest of the name for you or present choices for you to choose.
- 5. At Window menu, choose Show View. There are different views for looking at your project. We will introduce 2 of them here:
  - 5.1. package explorer shows only parts related to writing programs.
  - 5.2. navigator shows everything in the project directory.

6. If you want to change a variable name or file name, use **Refactor**. It is better than search and replace because it updates all relationships with other project components for you. To use refactor, select project name, class name, or anything that you want to rename, and right-click then select **Refactor -> Rename** as shown in Figure 16.



Figure 16 - Use Refactor to rename everything

- 7. When you are trying to run a program, make sure you are focusing on its file.
- 8. To run the program you run recently, click on
- 9. When you are deleting a project, you can choose to delete content (delete all the files in project folder). If you do not choose to delete content, Eclipse simply makes the project invisible (the real files still exist).

## Java review

#### **Console Keyboard Input**

### **Keyboard Input Dialog**

### showInputDialog

```
public static <u>String</u> showInputDialog(<u>Object</u> message,
<u>Object</u> initialSelectionValue)
```

Shows a question-message dialog requesting input from the user, with the input value initialized to initialSelectionValue. The dialog uses the default frame, which usually means it is centered on the screen.

#### Parameters:

message - the object to display initialSelectionValue - the value used to initialize the input field

```
import javax.swing.JOptionPane;
public static void main(String[] args) {
    // use showInputDialog to get input
    String text = JOptionPane.showInputDialog("Enter your input: ");
```

# }

## Lab 1 Exercise



### Your turn

#### NOTE:

- This is the first lab. Therefore it will be MUCH easier than future labs. Do not take it for granted.
- Prepare yourself by review material from previous semester.
- Be on time.
- You can help each other, but don't copy.
- Do not hesitate to ask teacher and TA. (Try to solve the problem by yourself first.)
- Use Java's help from java.sun.com or j2se6.chm.
- When you finish each exercise, you will ask a lab's instructor to verify the results and sign your lab. results' form. You should create a new project and new folder for each exercise.
- 1. Write a Java program that prints to the screen all prime numbers from 2 to 1000, ten numbers per line.
- 2. Write a Java program that prints out all elements of a one-dimension array of int. For example, if an array of int has 3, 5, 2, 6, 3 as its elements, the program should print { 3, 5, 2, 6, 3 }
- 3. Write a Java program that read 2 strings. The program will print "True" to the screen if all characters in the first string are in the second string (regardless of case), print "False" otherwise.

For example, if the first string is "*This is it.*" and the second string is "*It is the answer*", the program will print "*True*". However, if the first string is "*It is the answer*" and the second string is "*This is it*", the program will print "*False*", since there is no '*a*' or '*w*' or '*r*' in the second string.



Chulalongkorn University International School of Engineering Department of Computer Engineering

Name	
Student ID.	
Station No.	
Data	

## Lab 1 – Introduction to Eclipse.

Task	Description	Result	Note
1	Create workspace		
2	Create a new Java Project		
3	Import <ul> <li>.jar (or .zip) file</li> <li>Unzipped file(s)</li> </ul>		
4	Print prime numbers		
5	Print array elements		
6	Compare two Strings		
7			