

Engi- 5895 Lab 1

Hafez Seliem

Faculty of Engineering & Applied Science Memorial University of Newfoundland

What's Eclipse?

- It is a free software / open source platform-independent software framework for delivering what the project calls "richclient applications". Eclipse is also a community of users, constantly extending the covered application areas.
- Eclipse was originally developed by IBM as the successor of its VisualAge family of tools.
- Eclipse is now managed by the Eclipse Foundation, an independent not-for-profit consortium of software industry vendors.

Installing Eclipse

- Download free of charge from www.eclipse.org
- Eclipse is installed by default in all of the oncampus computers.

Installing eclipse

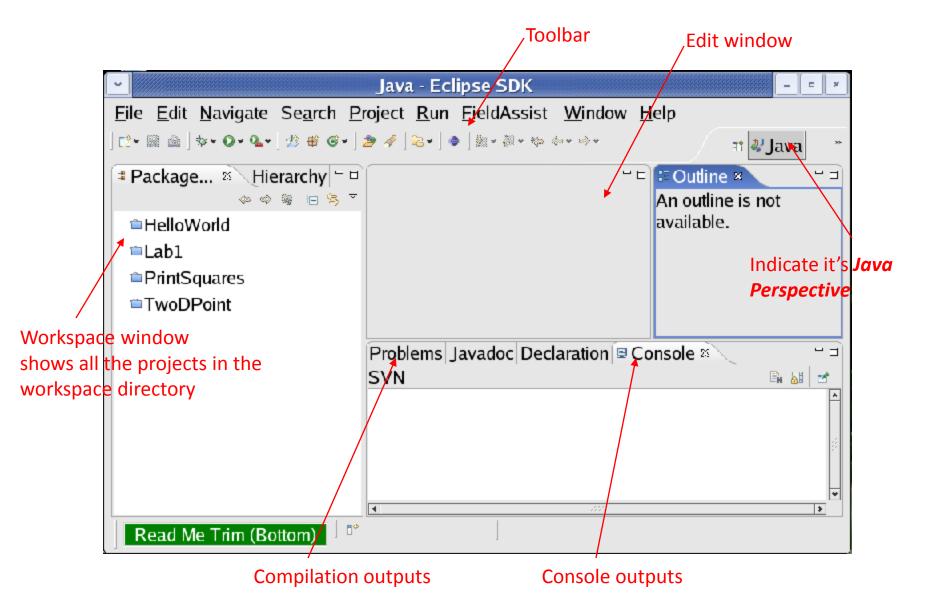
- Note: You can skip this step on the school computer, since Java and eclipse are already installed on it
- Prerequisite for eclipse: You need the Java Runtime Environment (JRE) in order to use eclipse
 - More than likely the JRE is already installed on your computer
 - If not, click on this link and follow the instructions:
 <u>Download JRE</u>

Eclipse start

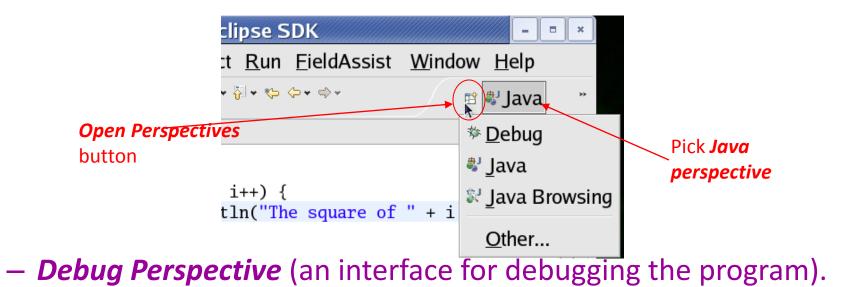
• Step 1: Open Eclipse, choose your workspace



Overview of Eclipse Java Perspective



- Choose a perspective (the layout of Eclipse user interface).
 - Open *Java perspective* (an interface for editing java source code): click *Open Perspective* button > click *Java*.



Create project

- On the Menu select "File / New / Project ...".
- Select "Java Project" and click "Next".
- Set "Project Name" to "Calculator"
- Set the execution environment to JavaSE-1.8.
- Click Finish.

💓 New Java Project					
Create a Java Project Create a Java project in the workspace or in an external location.					
Project name: Calculator					
Use default location					
Location: C:\Users\Hafez\workspace\Calcul	ator Browse				
JRE					
Our Sean execution environment JRE:	JavaSE-1.8				
Use a project specific JRE:	jre1.8.0_144 v				
O Use default JRE (currently 'jre1.8.0_144')	Configure JREs				
Project layout O Use project folder as root for sources and class files O Create separate folders for sources and class files Configure default					
Working sets					
Add project to working sets					
Working sets:	→ Select				
Reck	ext > Finish Cancel				

Create a class

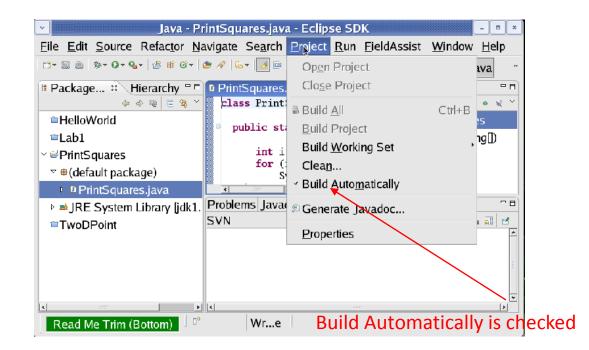
- Right-click on the Calculator project
- On the context menu select "New / Class
- Set name "CaclApp"
- Select public static

Void main

Source folder:	Calculator1/src	Browse
Package:	(default)	Browse
Enclosing type:		Browse
Name:	CalcApp	
Modifiers:	public package private protected abstract final static	
Superclass:	java.lang.Object	Browse
Interfaces:		Add Remove
Which method stu	bs would you like to create?	
	v public static void main(String[] args)	
	Constructors from superclass	
Do you want to add	☑ Inherited abstract methods d comments? (Configure templates and default value <u>here</u>) ☑ Generate comments	

Compile the program

• If *Build Automatically* is checked, the program will be automatically compiled whenever you save the program.



Run the program

- In the Package Explorer, right click on the file that has the main function.
- On the popup context menu, select 'Run As / Java Application'

	8 8 7				
🔛 awt-example					
Calculator					
B B src B B calculator	New		•		
🖲 🚺 DigitListi	Open UML		+		
🛞 🚺 Model.ja	Open	F3			
Operation	Open With		- E		
· View.jay	Open Type Hierarchy	F4			
calculate	Sho <u>w</u> In	Alt+Shift+W	•		
	Copy	Ctrl+C			
	Copy Qualified Name				
	Paste .	Ctrl+V			
 Calculator.uml CalculatorLab.ur 	💢 Delete	Delete			
	3b. Ramove from Context	C01+Alt+Shift+Doim			
Chess	Build Path				
FancyCalculator	Source	Alt+Shift+S	- E		
Helo mouse	Refactor	Alt+Shift+T	F		
Schematic	import				
sorting-with-observe	Copget				
References Declarations Refresh Assign Working Sets Bun As					
	Declarations		• 😥 Dedaratio	an	
	🔅 Refresh	F5	10	105	
	åssign Working Sets			Resource	Path
			• DI Java Ac	opication A	It+Shift+X, J
	Debug As Validate		O Open Rug	Dislaw	

Try your first java program

- Simple-command-line-calculator
 - Arithmetic functions: +, -, /, *.
 - Ask user to enter two number
 - Ask user to enter the operation
 - Display the result for the user
 - After you finish, improve your code by separating the main function from the model code (create "Operation" class that has (add, sub, multiply, sum) methods).

More to try

- Simple-command-line-calculator
 - Arithmetic functions: add, sub, mult, div, each taking two arbitrary expressions as arguments.
 - For example

Input	Output
add(1, 2)	3
add(1, mult(2, 3))	7
<pre>mult(add(2, 2), div(9, 3))</pre>	12