

IDE

What Is an Integrated Development Environment (IDE)?

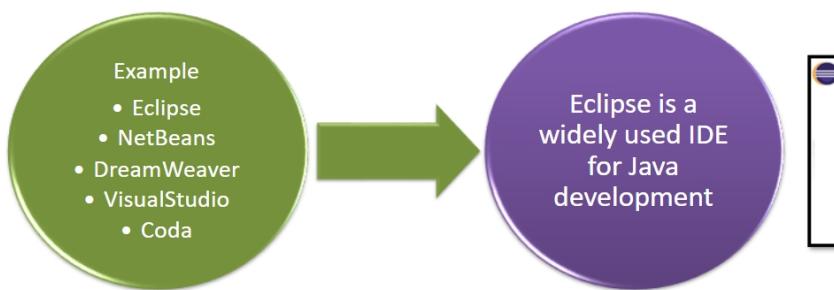
An integrated development environment (IDE) is an application that facilitates application development with ease.

IDE will consist of

- Source Code Editor
- Compiler
- Debugger
- Build Automation Tools

IDE may support either a single language or multiple language

IDE



Introduction to Eclipse

Eclipse is free to download and install

It is composed of plugins.

Few of those plugins

- Java Development Tools (JDT) for Java
- PyDev for Python
- C / C++ Development Tools (CDT) for C / C++
- PHPeclipse for PHP

Eclipse platform and other plug-ins from the Eclipse foundation is released under the Eclipse Public License (EPL).

Every year new versions of Eclipse are released

Installation and Setting up of Eclipse



Different versions of eclipse are available

Download eclipse from <http://www.eclipse.org/downloads/>

Choose eclipse version based on the operating system

First download the zip file and extract it.

Eclipse is now ready to start

Start the eclipse by clicking the eclipse.exe file



Create a workspace

Eclipse prompts for workspace

- Workspace is the location where multiple projects are created and stored
- The *workspace* is the physical location (file path) for storing certain meta-data and the development artifacts.

If the workspace does not exist, it creates one

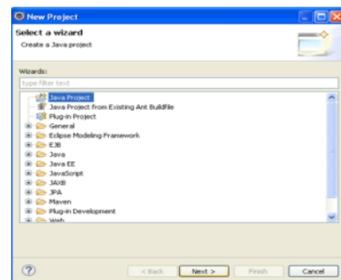
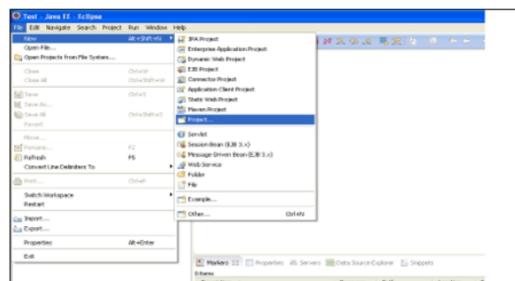


Create Java Project

Select File -> New -> Java project from the menu

Enter the Project Name and Click Finish

A Java Project is created and opened in Java Perspective



Create a package

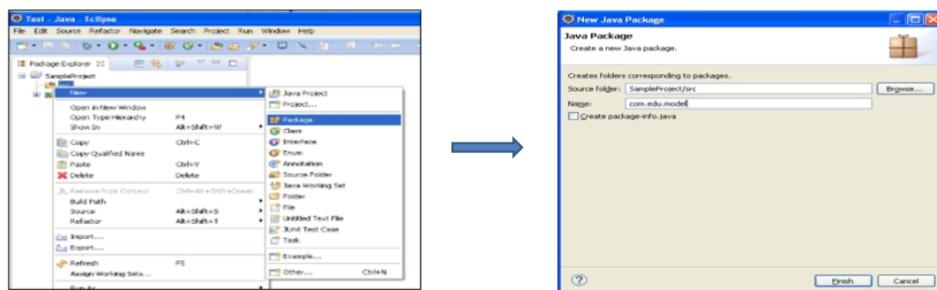


A package is a namespace that groups related classes

Create a new package

- Right click the "src" folder in the project -> New -> Package

Enter the package name and press Finish



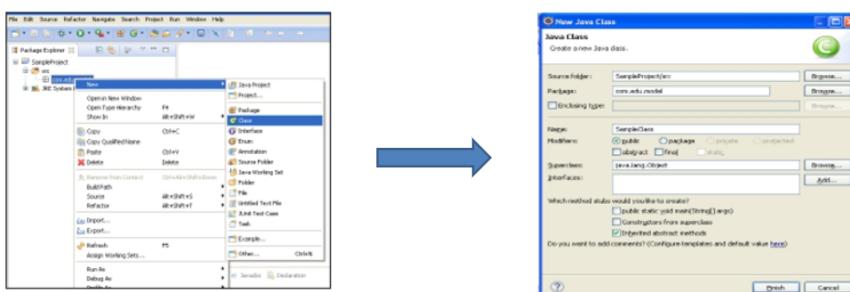
Create a class



Right click the package -> New -> Class

Enter the class name, say SampleClass

Select the `public static void main (String[] args)` checkbox.(optional)



Create a class



```
Test - Java - SampleProject/src/com/edu/model/SampleClass.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer SampleProject
  src
    com.edu.model
      SampleClass.java
JRE System Library [JavaSE-1.8]

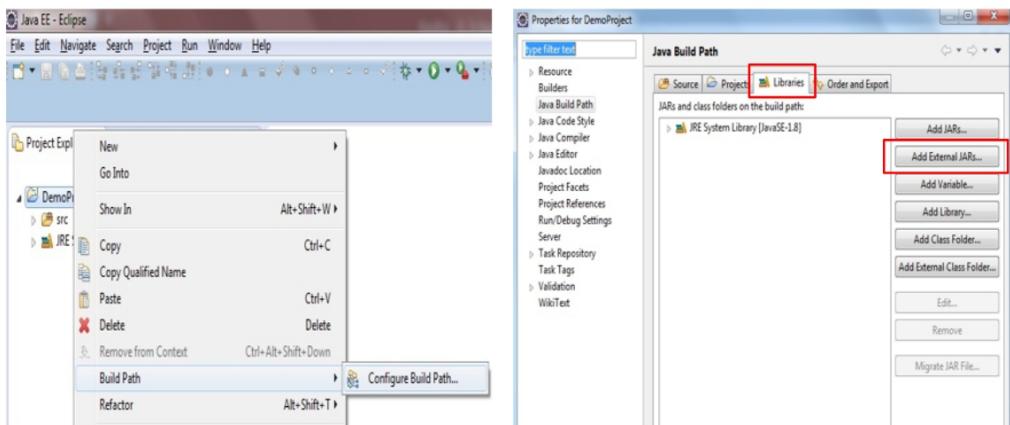
SampleClass.java
1 package com.edu.model;
2
3 public class SampleClass {
4     public static void main(String args[]){
5         System.out.println("HelloWorld");
6     }
7
8 }
9
```

IMPORT A JAR FILE

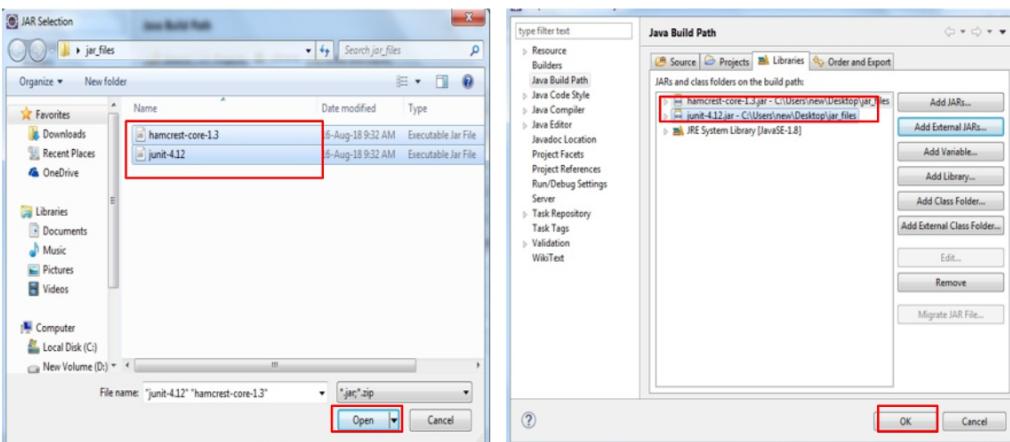


- 👉 Right click on project => Select Build Path => Configure Build path
- 👉 The "Java Build Path" box pops up
- 👉 Select Libraries and select Add External JARs
- 👉 Select JAR files and click ok

IMPORT A JAR FILE



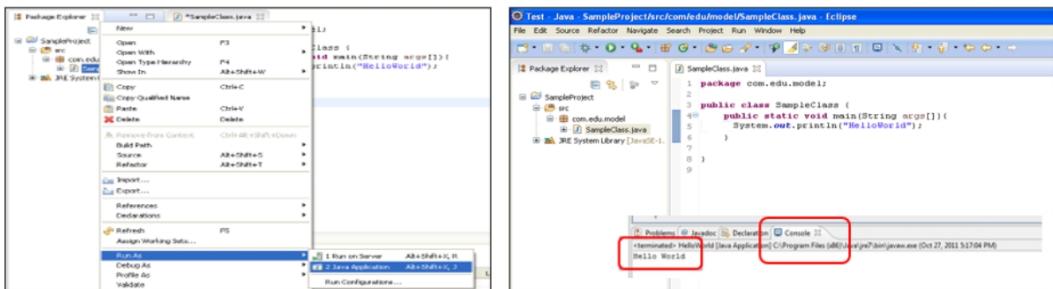
IMPORT A JAR FILE



Execute the Program

Right click the class file -> Run As -> Java Application

- Output is seen in the console view

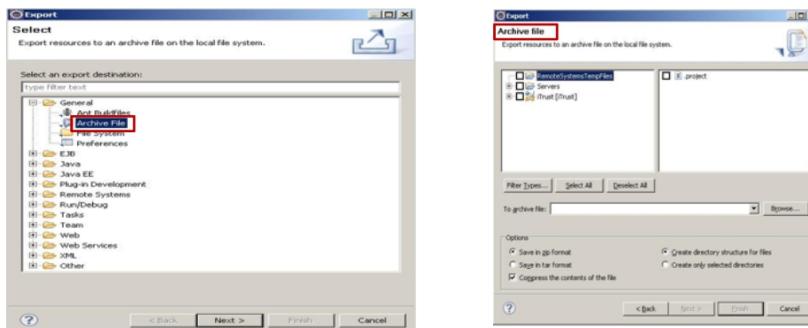


Also the class can be executed by clicking the Run icon  on the toolbar

Miscellaneous Options – Export Project

Can Export an entire Eclipse project into an archive file,

- Go to File -> Export.



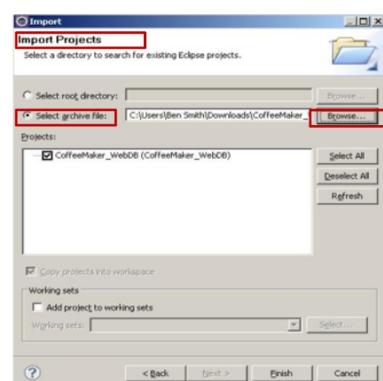
Click Next

Click **Finish**. Eclipse will automatically export and zip the project into an archive file and store it in the location specified.

Miscellaneous Options – Import Project

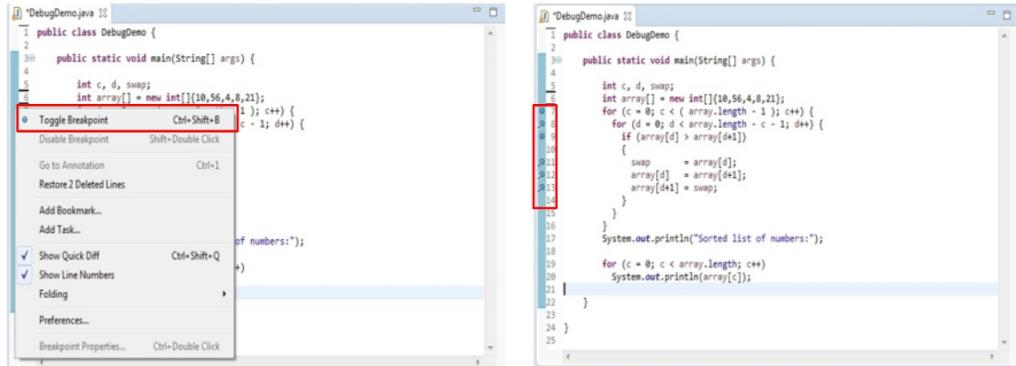
Can Import an entire Eclipse project from an archive file

- Go to File -> Import.



- Select **Existing Projects into Workspace**.
- Click the radio button next to **Select archive file** and click the **Browse** button
- Choose the file path for the zip file
- Click **finish**

DEBUGGING USING ECLIPSE



```

1 public class DebugDemo {
2     public static void main(String[] args) {
3         int c, d, swap;
4         int array[] = new int[]{10,56,4,8,21};
5         for (c = 0; c < array.length - 1; c++) {
6             for (d = c + 1; d < array.length - 1; d++) {
7                 if (array[d] > array[d+1]) {
8                     swap = array[d];
9                     array[d] = array[d+1];
10                    array[d+1] = swap;
11                }
12            }
13        }
14        System.out.println("Sorted list of numbers:");
15        for (c = 0; c < array.length; c++)
16            System.out.println(array[c]);
17    }
18 }
19
20
21
22
23
24
25

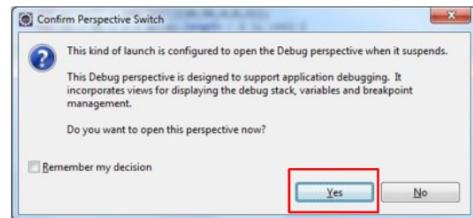
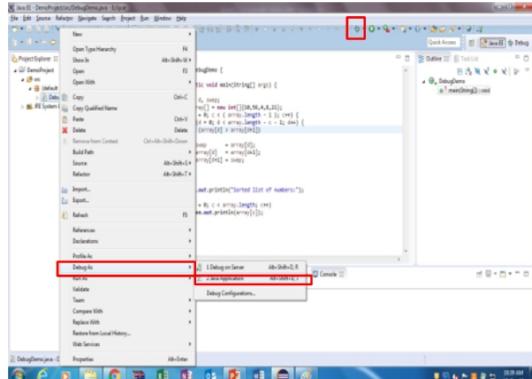
```

Run a java program and understand source code and the values of variables while execution.

Breakpoints in the source code helps to investigate the values of variable.

Right-click the left margin in the Java editor and select **Toggle Breakpoint**

DEBUGGING USING ECLIPSE



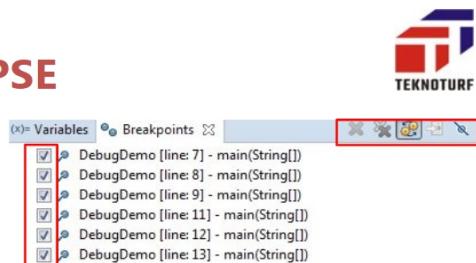
To debug the java program Select the file with main method

Right-click on it and select **Debug As** => **Java Application** (or) Use **Debug icon**

on the 

DEBUGGING USING ECLIPSE

Toolbars	Keyboard Shortcut	Action
	F8	Resume
	Ctrl + F2	Terminate
	F5	Step Into
	F6	Step Over
	F7	Step Return



Name	Value
args	String[0] (id=16)
array	(id=17)
c	0
d	1
swap	56

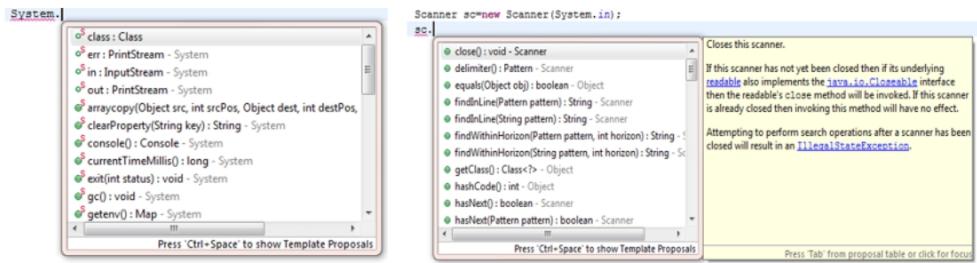
Miscellaneous Options – Code Assist



When typing the code in the text editor, eclipse provides code assist, termed as Intellisense.

Code Assist is

- Either triggered automatically (when we type a dot after variable or class name)
- Or forced by pressing Ctrl + Space on code



Miscellaneous Options – Eclipse Shortcuts



Eclipse IDE has many shortcuts that help in fast development

Few shortcuts that are often used:

Shortcut	Description
CTRL SPACE	Type assist
CTRL SHIFT F	Format code
CTRL /	Comment a line
F3	Go to the declaration of the variable
F4	Show type hierarchy of a class
ALT SHIFT Z	Enclose block in try-catch
CTRL F11	Run last run program
CTRL 1	Quick fix code

Miscellaneous Options – Organize Imports

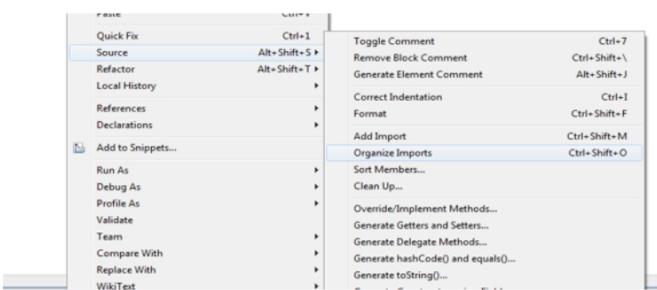


Using Eclipse IDE, all the classes needed for a file can be imported automatically

Right click -> Source -> Organize Imports

This will import the class files needed

If unwanted class files are imported, those will be deleted



Miscellaneous Options – Auto Format



Eclipse provides auto format feature for the code

To make a code readable and maintainable , it should be properly aligned and indented

In Eclipse, press **Ctrl+Shift+F** on code to format it.

- If a portion of code is selected, it formats the selection.
- Else it formats the entire source file.

```
public class TestDemo {  
    public void add(int a,int b)  
    {  
        int c; c=a+b; int d; d=a-b;  
        int x=a*b;  
    }  
}
```

Ctrl + Shift + F

```
public class TestDemo {  
    public void add(int a, int b) {  
        int c;  
        c = a + b;  
        int d;  
        d = a - b;  
        int x = a * b;  
    }  
}
```

Miscellaneous Options – Fix Compilation Errors



In Eclipse, the code editor shows compilation error, if any, as the code is typed

IDE provides suggestion on how to fix the compilation error

```
public long add(long a,long b)  
{  
    int c=a+b;  
}
```

```
public long add(long a,long b)  
{  
    Type mismatch: cannot convert from long to int  
}
```