MALLA REDDY ENGINEERING COLLEGE
(Autonomous)

Maisammaguda, Dhulapally (post & viaKompally), Secunderabad-500100.

Department of Computer Science & Information Technology

OBJECT ORIENTED PROGRAMMING THROUGH JAVA LAB

SUBJECT CODE: C0511

B.Tech -III Semester (MR23)

ACADEMIC YEAR: 2024-2025

Department of Computer Science & Information Technology

INSTITUTION VISION

To be a premier center of professional education and research, offering quality programs in a socio-economic and ethical ambience.

INSTITUTION MISSION

- To impart knowledge of advanced technologies using state-of-the-art infrastructural facilities.
- To inculcate innovation and best practices in education, training and research.
- To meet changing socio-economic needs in an ethical ambience.

Malla Reddy Engineering College

(UGC Autonomous Institution, Approved by AICTE, New Delhi & Affiliated to JNTUH,
Hyderabad and Accredited 3rd cycle by NAAC with ‘A++’ Grade)
Maisammaguda(H),Medchal-MalkajgiriDistrict,Secunderabad,TelanganaState– 500100,
www.mrec.ac.in

Department of Computer Science & Information Technology

DEPARTMENT VISION

To Attain Global Standards in the Teaching, Training, and Research of the IT Industry that Strike a Balance between the Rising Needs of the Sector and the Socio-Economic and Ethical Needs of the Society.

DEPARTMENT MISSION

M1: To Impart Quality Education and Research to the Students in Information Technology.

M2: To Train the Students in Advanced Technologies using State-of-the-Art Facilities.

M3: To Build the Knowledge, Skills and Aptitude to Succeed in the Information Technology Industry through Ethical Values and Social Relevance.

.

Malla Reddy Engineering College

(UGC Autonomous Institution, Approved by AICTE, New Delhi & Affiliated to JNTUH,
Hyderabad and Accredited 3rd cycle by NAAC with ‘A++’ Grade)
Maisammaguda(H),Medchal-MalkajgiriDistrict,Secunderabad,TelanganaState– 500100,
www.mrec.ac.in

Department of Computer Science & Information Technology

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

PEO1: To outshine in professional career with sound problem solving ability for providing IT solutions by proper plan, analysis, design, implementation and validation.

PEO2: To pursue training, advance study and research using scientific, technical and communication base to cope with the evolution in the technology.

PEO3: To utilize the acquired technical skills and knowledge for the benefit of society

PROGRAMME SPECIFIC OUTCOMES (PSOs)

PSO1: Identify the mathematical abstractions and algorithm design techniques together with emerging Software Tools to solve complexities indulged in efficient programming.

PSO2: Apply the core concepts of current technologies in the hardware, software domains inaccomplishing IT enabled services to meet out societal needs.

PSO3: Practice modern computing techniques by continual learning process with ethical concerns in establishing innovative career path.

Malla Reddy Engineering College

(UGC Autonomous Institution, Approved by AICTE, New Delhi & Affiliated to JNTUH,
Hyderabad and Accredited 3rd cycle by NAAC with ‘A++’ Grade)
Maisammaguda(H),Medchal-MalkajgiriDistrict,Secunderabad,TelanganaState– 500100,
www.mrec.ac.in

Department of Computer Science & Information Technology

PROGRAMME OUTCOMES (POs)

PO 1	**Engineering knowledge**: Apply the knowledge of mathematics, science, engineering Fundamentals and an engineering specialization to the solution of complex engineering problems.
PO 2	**Problem analysis**: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO 3	**Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, andenvironmental considerations.
PO 4	**Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis ofthe information to provide valid conclusions.
PO 5	**Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern Engineering and IT tools including prediction and modeling to complex engineering activitieswith an understanding of the limitations.
PO 6	**The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevantto the professional engineering practice.
PO 7	**Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledgeof, and need for sustainable development.
PO 8	**Ethics**: Apply ethical principles and commit to professional ethics and responsibilities andnorms of the engineering practice.
PO 9	**Individual and team work**: Function effectively as an individual and as a member or leaderin diverse teams, and in multidisciplinary settings.
PO 10	**Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend andwrite effective reports and design documentation, make effective presentations, and give and receive clear instructions.

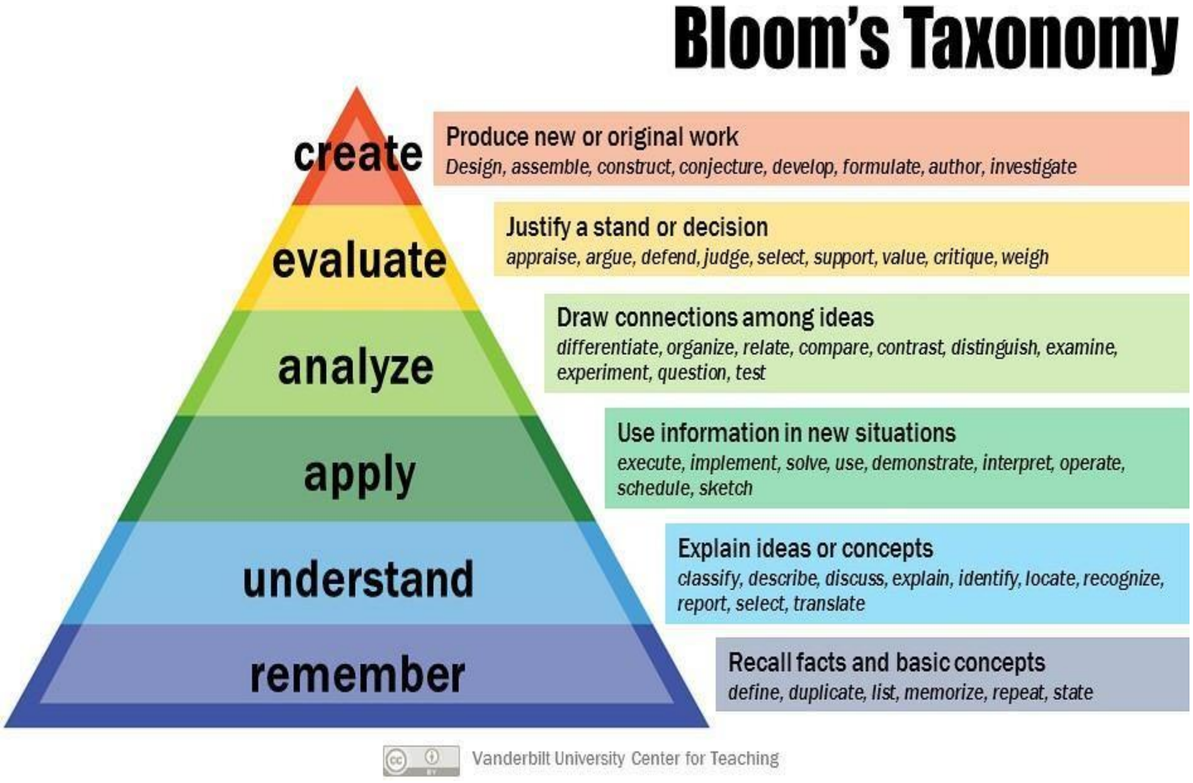
	PO 11	**Project management and finance**: Demonstrate knowledge and understanding of the Engineering and management principles and apply these to one’s own work, as a member andleader in a team, to manage projects and in multidisciplinary environments.
	PO 12	**Life-long learning**: Recognize the need for, and have the preparation and ability to engage inindependent and life-long learning in the broadest context of technological change.

Malla Reddy Engineering College

(UGC Autonomous Institution, Approved by AICTE, New Delhi & Affiliated to JNTUH,
Hyderabad and Accredited 3rd cycle by NAAC with 'A++' Grade)
Maisammaguda(H),Medchal-MalkajgiriDistrict,Secunderabad,TelanganaState– 500100,
www.mrec.ac.in

Department of Computer Science & Information Technology

Bloom's Taxonomy Triangle



Bloom's Taxonomy

create — Produce new or original work
Design, assemble, construct, conjecture, develop, formulate, author, investigate

evaluate — Justify a stand or decision
appraise, argue, defend, judge, select, support, value, critique, weigh

analyze — Draw connections among ideas
differentiate, organize, relate, compare, contrast, distinguish, examine, experiment, question, test

apply — Use information in new situations
execute, implement, solve, use, demonstrate, interpret, operate, schedule, sketch

understand — Explain ideas or concepts
classify, describe, discuss, explain, identify, locate, recognize, report, select, translate

remember — Recall facts and basic concepts
define, duplicate, list, memorize, repeat, state

Vanderbilt University Center for Teaching

2024-25 Onwards (MR23)	**Malla Reddy Engineering College (Autonomous)**	**B.Tech. III Semester**		
Code: B0511	**Object Oriented Programming through Java Lab (Common for CSE, IT, CSIT, CSE(DS), CSE(AIML), CSE(CS), and CSE (IOT))**	**L**	**T**	**P**
Credits: 1.5		**-**	**-**	**3**

Prerequisites: NIL

Course Objectives:
This course will make students able to learn and understand the concepts and features of object- oriented programming and the object-oriented concept like inheritance and will know how to make use of interfaces and package, to acquire the knowledge in Java's exception handling mechanism, multithreading, to explore concepts of Applets and event handling mechanism. This course makes students to gain the knowledge in programming using Layout Manager and swings.

- ***Software Requirements:*** *Java*

List of Programs

1. Write Java Programs that implement the following.
   a) Constructor
   b) Parameterized constructor
   c) Method Overloading
   d) Constructor overloading
2. Write a Java program
   a) checks whether a given string is a palindrome or not.
   b) for sorting a given list of names in ascending order.
   c) that reads a line if integers and then displays each integer and the sum of all integers(use string tokenizer class of java.util).
3. Write Java programs that uses the following keywords…
   a) this
   b) super
   c) static
   d) final
4. Write a Java program to implement
   a) Method Overriding.
   b) dynamic method dispatch.
   c) multiple inheritance.
   d) access specifiers.
5. Write a Java program that
   a) reads a file name from the user, and then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes.
   b) reads a file and displays the file on the screen, with a line number before each line.
   c) displays the number of characters, lines and words in a test file.
6. Write a Java program for handling
   a) Checked exceptions.
   b) unchecked exceptions.
7. Write a Java program
   a) Creates three threads. First threads displays "Good Morning "for every one second, the second thread displays "Hello" for every two seconds, the third thread Displays "Welcome" for every three seconds.

b) that correctly implements producer consumer problem using concept of inter thread communication.

8. Write a Java program which demonstrates the use of following collection classes
   a) Array List
   b) Hash Set
   c) Deque
9. Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +,-,*,/ operations. Add a text field to displaythe result.
10. Write a Java program for handling
    a) mouse events.
    b) key events.
11. Write a Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields num1 and num2. The division of num1 and num2 is displayed in the result field when the divide button is clicked. If num1 or num2 were not an integer, the program would throw number format exception. If num2 were zero, the program would throw an arithmetic exception and display the exception in the message dialogue box.

12. Write a Java program that
    a) Simulates traffic light. The program lets the user select one of three lights: red, yellow or green. When a radio button is selected, the light is turned on and only one light can be on at a time. No light is on when the program starts.
    b) Allows the user to draw lines rectangles and ovals.

TEXT BOOKS:

1. Herbert Schildt, “**Java The complete reference**”, TMH, 8th edition, 2011.
2. T. Budd, “**Understanding OOP with Java**”, Pearson Education, updated edition, 1998.

REFERENCES:

1. P.J. Deitel and H.M. Deitel, “**Java for Programmers**”, Pearson education.
2. P. Radha Krishna, “**Object Oriented Programming through Java**”, Universities Press.
3. Bruce Eckel,” **Programming in Java**”, Pearson Education.
4. S. Malhotra and S. Choudhary,” **Programming in Java**”, Oxford Univ. Press.

Course Outcomes:

At the end of the course, students will be able to

1. **Build** simple java programs using the basic concepts of OOP
2. **Create** user defined packages to build real time applications
3. **Develop** applications on files, exceptions, threads and applets.
4. **Construct** GUI based applications.
5. **Design** Interactive applications for use on internet.

CO- PO, PSO Mapping **(3/2/1 indicates strength of correlation) 3-Strong, 2-Medium, 1-Weak**															
COs	**Programme Outcomes (POs)**												**PSOs**		
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**
CO1	3	3	2	3	3	-	-	3	3	2	-	3	2	2	2
CO2	3	3	3	2	3	-	-	2	3	2	-	3	2	2	2
CO3	3	3	3	3	2	-	-	2	2	1	-	2	3	2	2
CO4	3	3	2	2	2	-	-	3	3	2	-	3	2	2	1
CO5	2	2	2	3	3	-	-	2	1	2	-	2	2	2	2

Department of AIML

1. Use Eclipse or Net bean platform and acquaint with the various menus. Create a test project, add a test class, and run it. See how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods, and classes. Try debug step by step with a small program of about 10 to 15 lines which contains at least one if else condition and a for loop.
2. Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -,*, % operations. Add a text field to display the result. Handle any possible exceptions like divided by zero.
3. **a)** Develop an applet in Java that displays a simple message.

   b) Develop an applet in Java that receives an integer in one text field, and computes its factorial Value and returns it in another text field, when the button named “Compute” is clicked.
4. Write a Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num 2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception. Display the exception in a message dialog box.
5. Write a Java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.
6. Write a Java program for the following: Create a doubly linked list of elements. Delete a given element from the above list. Display the contents of the list after deletion.
7. Write a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green with radio buttons. On selecting a button, an appropriate message with “Stop” or “Ready” or “Go” should appear above the buttons in selected color. Initially, there is no message shown.
8. Write a Java program to create an abstract class named Shape that contains two integers and an empty method named print Area (). Provide three classes named Rectangle, Triangle, and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.
9. Suppose that a table named Table.txt is stored in a text file. The first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas. Write a java program to display the table using Labels in Grid Layout.

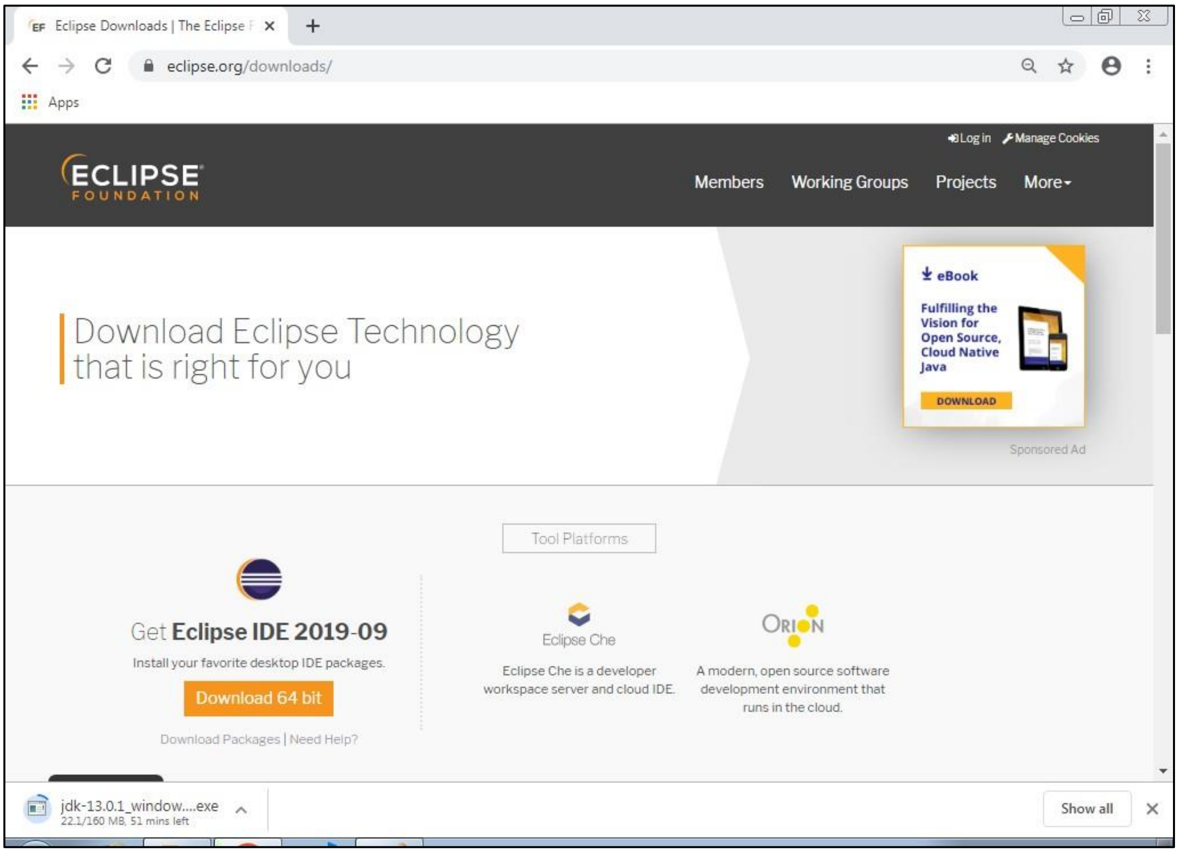
10. Write a Java program that handles all mouse events and shows the event name at the center of the window when a mouse event is fired (Use Adapter classes).
11. Write a Java program that loads names and phone numbers from a text file where the data is organized as one line per record and each field in a record are separated by a tab (\t). It takes a name or phone number as input and prints the corresponding other value from the hash table (hint: use hash tables).
12. Write a Java program that correctly implements the producer – consumer problem using the concept of interthread communication.
13. Write a Java program to list all the files in a directory including the files present in all its subdirectories.
14. Write a Java program that implements Quick sort algorithm for sorting a list of names in ascending Order.
15. Write a Java program that implements Bubble sort algorithm for sorting in descending order and also shows the number of interchanges occurred for the given set of integers.

Week 1.

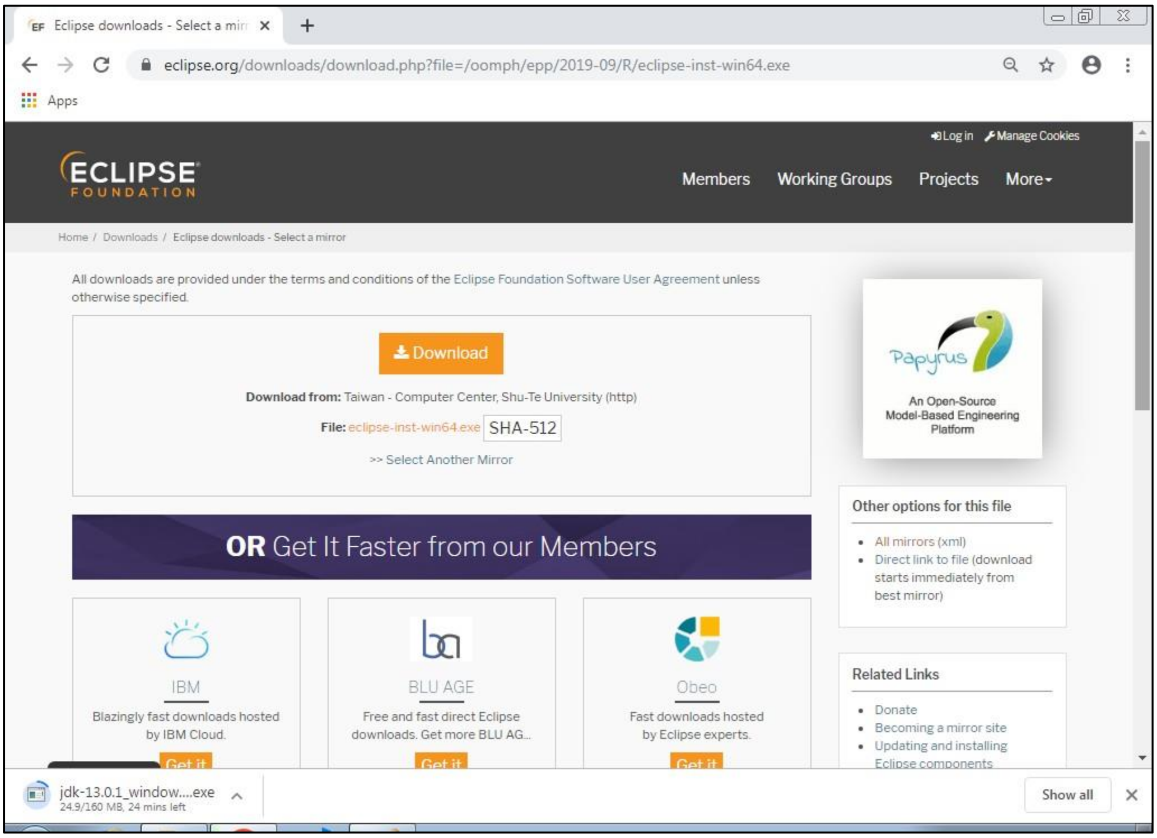
Aim: Use Eclipse or Net bean platform and acquaint with the various menus. Create a test project, add a test class, and run it. See how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods, and classes. Try debug step by step with a small program of about 10 to 15 lines which containsat least one if else condition and a for loop.

Solution:

- **Step 1 -** Install JDK in the computer.
- **Step 2 -** Set the path in the Environment Variables from Advanced Setting of computer
- **Step 3 -** Download Eclipse from Eclipse website
- **Step 4 -** Install the Eclipse (follow the screen to install eclipse)


Eclipse Downloads | The Eclipse F
eclipse.org/downloads/
Apps
Log in
Manage Cookies
ECLIPSE FOUNDATION
Members
Working Groups
Projects
More
Download Eclipse Technology that is right for you
eBook
Fulfilling the Vision for Open Source, Cloud Native Java
DOWNLOAD
Sponsored Ad
Tool Platforms
Get Eclipse IDE 2019-09
Install your favorite desktop IDE packages.
Download 64 bit
Download Packages | Need Help?
Eclipse Che
Eclipse Che is a developer workspace server and cloud IDE.
ORION
A modern, open source software development environment that runs in the cloud.
jdk-13.0.1_window....exe
22.1/160 MB, 51 mins left
Show all

Select the sultable version based on your OS.



Eclipse downloads - Select a mirr

eclipse.org/downloads/download.php?file=/oomph/epp/2019-09/R/eclipse-inst-win64.exe

Home / Downloads / Eclipse downloads - Select a mirror

All downloads are provided under the terms and conditions of the Eclipse Foundation Software User Agreement unless otherwise specified.

Download

Download from: Taiwan - Computer Center, Shu-Te University (http)

File: eclipse-inst-win64.exe SHA-512

>> Select Another Mirror

OR Get It Faster from our Members

IBM — Blazingly fast downloads hosted by IBM Cloud. Get it

BLU AGE — Free and fast direct Eclipse downloads. Get more BLU AG... Get it

Obeo — Fast downloads hosted by Eclipse experts. Get it

Papyrus — An Open-Source Model-Based Engineering Platform

Other options for this file

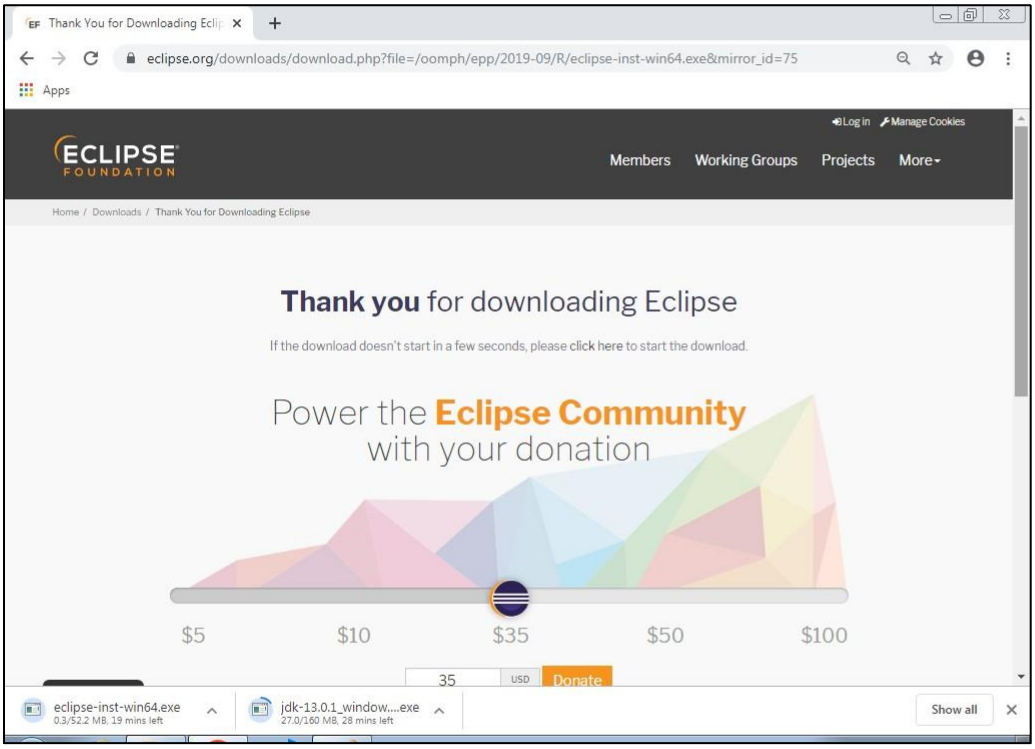
- All mirrors (xml)
- Direct link to file (download starts immediately from best mirror)

Related Links

- Donate
- Becoming a mirror site
- Updating and installing Eclipse components

jdk-13.0.1_window....exe 24.9/160 MB, 24 mins left

Then download get starts.



Thank You for Downloading Eclip

eclipse.org/downloads/download.php?file=/oomph/epp/2019-09/R/eclipse-inst-win64.exe&mirror_id=75

Home / Downloads / Thank You for Downloading Eclipse

Thank you for downloading Eclipse

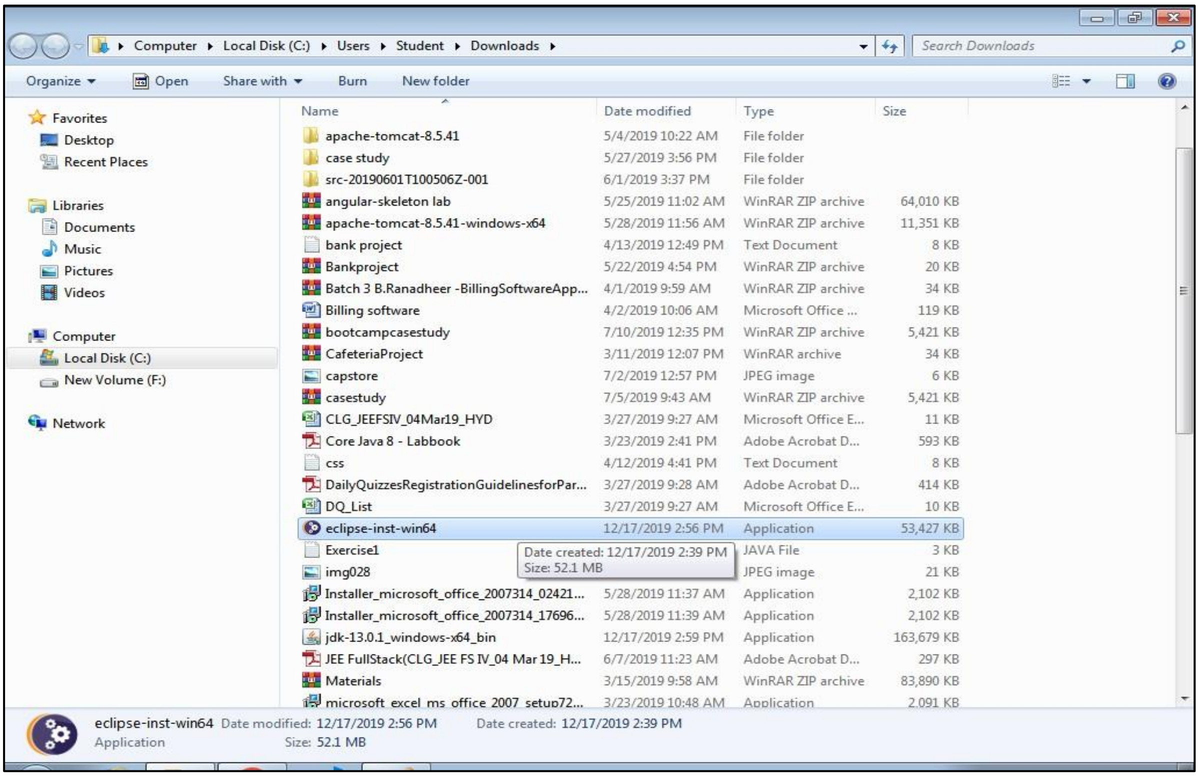
If the download doesn't start in a few seconds, please click here to start the download.

Power the **Eclipse Community** with your donation

eclipse-inst-win64.exe 0.3/52.2 MB, 19 mins left

jdk-13.0.1_window....exe 27.0/160 MB, 28 mins left

Double click on the Eclipse Application.

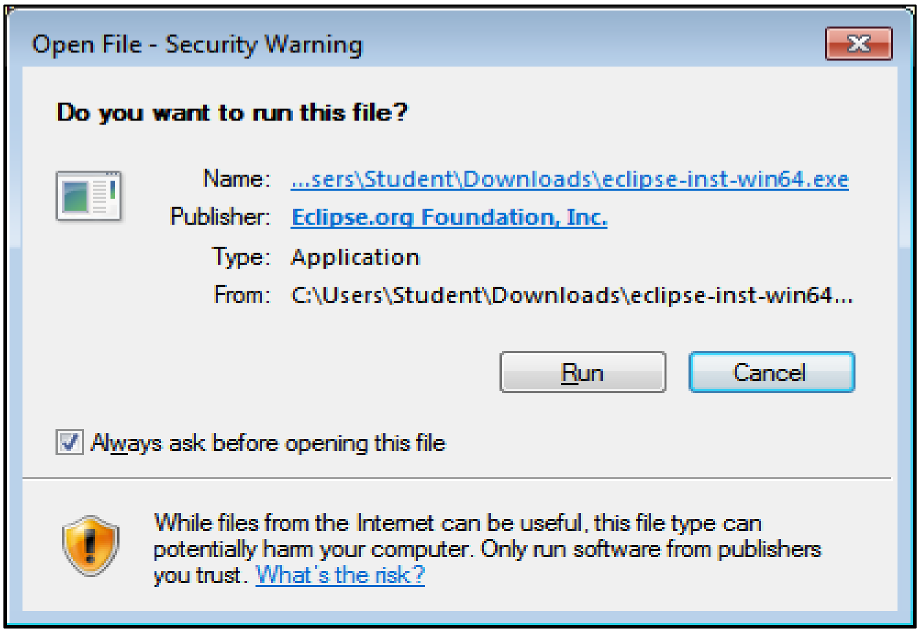


Name	Date modified	Type	Size
apache-tomcat-8.5.41	5/4/2019 10:22 AM	File folder	
case study	5/27/2019 3:56 PM	File folder	
src-20190601T100506Z-001	6/1/2019 3:37 PM	File folder	
angular-skeleton lab	5/25/2019 11:02 AM	WinRAR ZIP archive	64,010 KB
apache-tomcat-8.5.41-windows-x64	5/28/2019 11:56 AM	WinRAR ZIP archive	11,351 KB
bank project	4/13/2019 12:49 PM	Text Document	8 KB
Bankproject	5/22/2019 4:54 PM	WinRAR ZIP archive	20 KB
Batch 3 B.Ranadheer -BillingSoftwareApp...	4/1/2019 9:59 AM	WinRAR ZIP archive	34 KB
Billing software	4/2/2019 10:06 AM	Microsoft Office ...	119 KB
bootcampcasestudy	7/10/2019 12:35 PM	WinRAR ZIP archive	5,421 KB
CafeteriaProject	3/11/2019 12:07 PM	WinRAR archive	34 KB
capstore	7/2/2019 12:57 PM	JPEG image	6 KB
casestudy	7/5/2019 9:43 AM	WinRAR ZIP archive	5,421 KB
CLG_JEEFSIV_04Mar19_HYD	3/27/2019 9:27 AM	Microsoft Office E...	11 KB
Core Java 8 - Labbook	3/23/2019 2:41 PM	Adobe Acrobat D...	593 KB
css	4/12/2019 4:41 PM	Text Document	8 KB
DailyQuizzesRegistrationGuidelinesforPar...	3/27/2019 9:28 AM	Adobe Acrobat D...	414 KB
DQ_List	3/27/2019 9:27 AM	Microsoft Office E...	10 KB
eclipse-inst-win64	12/17/2019 2:56 PM	Application	53,427 KB
Exercise1		JAVA File	3 KB
img028		JPEG image	21 KB
Installer_microsoft_office_2007314_02421...	5/28/2019 11:37 AM	Application	2,102 KB
Installer_microsoft_office_2007314_17696...	5/28/2019 11:39 AM	Application	2,102 KB
jdk-13.0.1_windows-x64_bin	12/17/2019 2:59 PM	Application	163,679 KB
JEE FullStack(CLG_JEE FS IV_04 Mar 19_H...	6/7/2019 11:23 AM	Adobe Acrobat D...	297 KB
Materials	3/15/2019 9:58 AM	WinRAR ZIP archive	83,890 KB
microsoft_excel_ms_office_2007_setup72...	3/23/2019 10:48 AM	Application	2,091 KB

Date created: 12/17/2019 2:39 PM
Size: 52.1 MB

eclipse-inst-win64 Date modified: 12/17/2019 2:56 PM Date created: 12/17/2019 2:39 PM
Application Size: 52.1 MB

Click on Run in the Security Warning box.



Open File - Security Warning

Do you want to run this file?

Name: ...sers\Student\Downloads\eclipse-inst-win64.exe

Publisher: Eclipse.org Foundation, Inc.

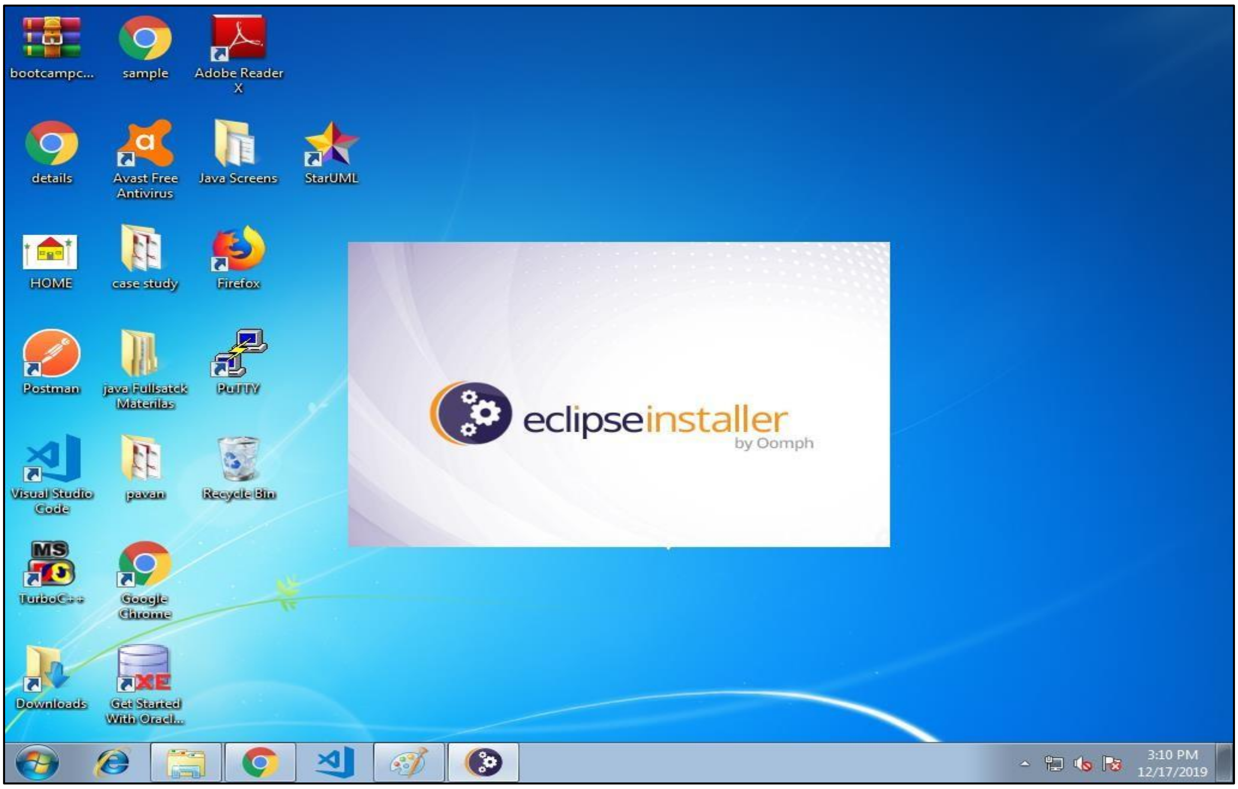
Type: Application

From: C:\Users\Student\Downloads\eclipse-inst-win64...

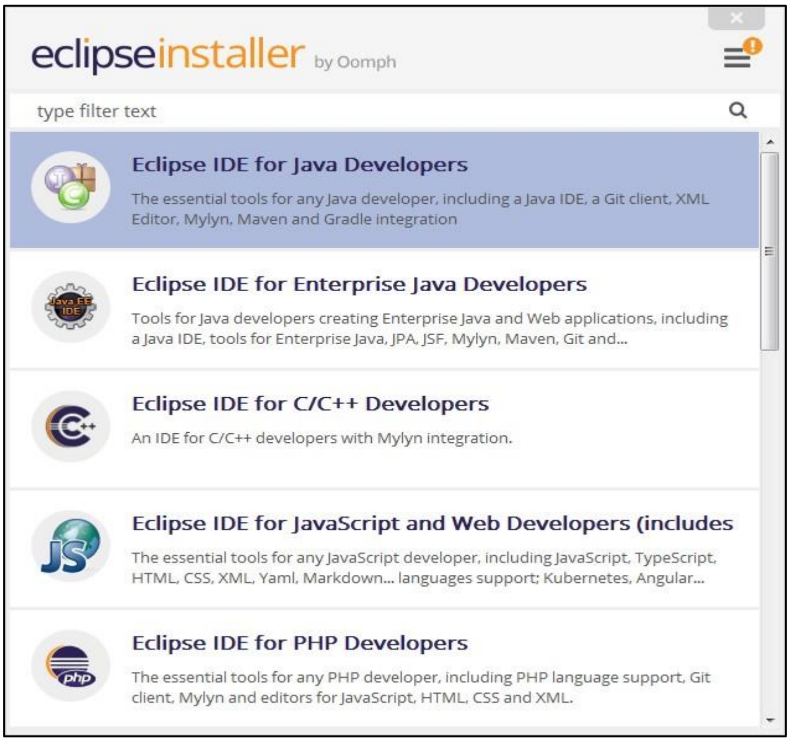
☑ Always ask before opening this file

While files from the Internet can be useful, this file type can potentially harm your computer. Only run software from publishers you trust. What's the risk?

Then, the installation process begins.



Click on Eclipse IDE for Java Developers.


eclipseinstaller by Oomph

type filter text

Eclipse IDE for Java Developers
The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven and Gradle integration

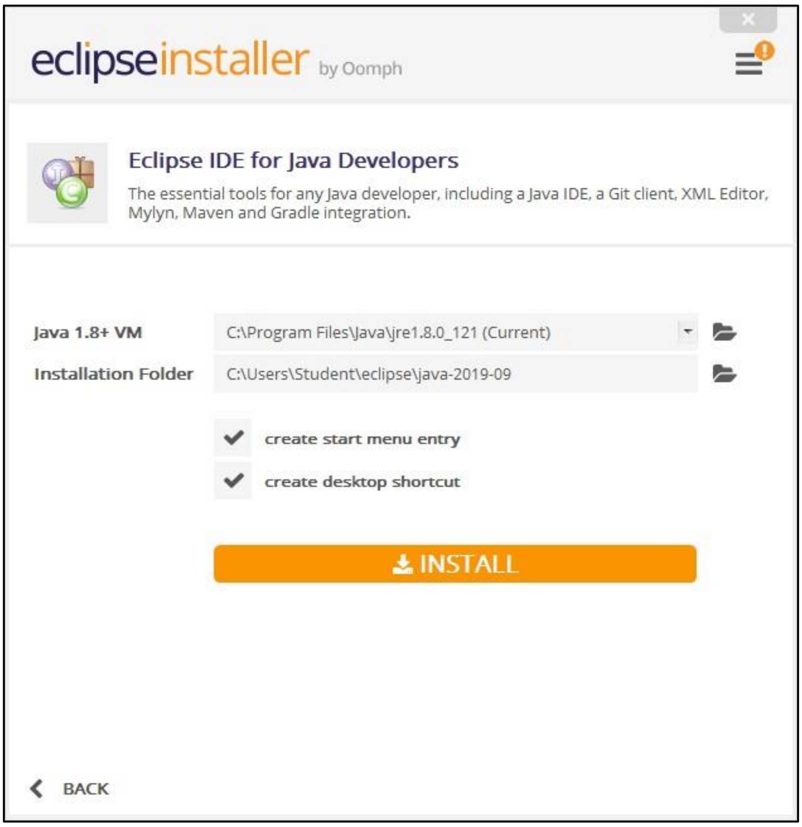
Eclipse IDE for Enterprise Java Developers
Tools for Java developers creating Enterprise Java and Web applications, including a Java IDE, tools for Enterprise Java, JPA, JSF, Mylyn, Maven, Git and...

Eclipse IDE for C/C++ Developers
An IDE for C/C++ developers with Mylyn integration.

Eclipse IDE for JavaScript and Web Developers (includes
The essential tools for any JavaScript developer, including JavaScript, TypeScript, HTML, CSS, XML, Yaml, Markdown... languages support; Kubernetes, Angular...

Eclipse IDE for PHP Developers
The essential tools for any PHP developer, including PHP language support, Git client, Mylyn and editors for JavaScript, HTML, CSS and XML.

Click on Install button.



Eclipse IDE for Java Developers

The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven and Gradle integration.

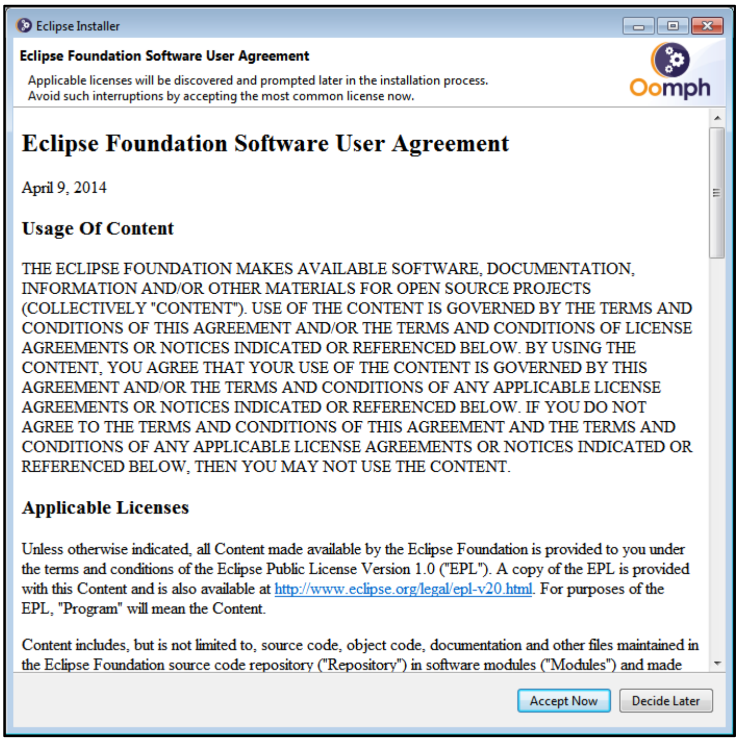
Java 1.8+ VM: C:\Program Files\Java\jre1.8.0_121 (Current)

Installation Folder: C:\Users\Student\eclipse\java-2019-09

☑ create start menu entry

☑ create desktop shortcut

Click on Accept Now.



Eclipse Foundation Software User Agreement

Applicable licenses will be discovered and prompted later in the installation process.
Avoid such interruptions by accepting the most common license now.

Eclipse Foundation Software User Agreement

April 9, 2014

Usage Of Content

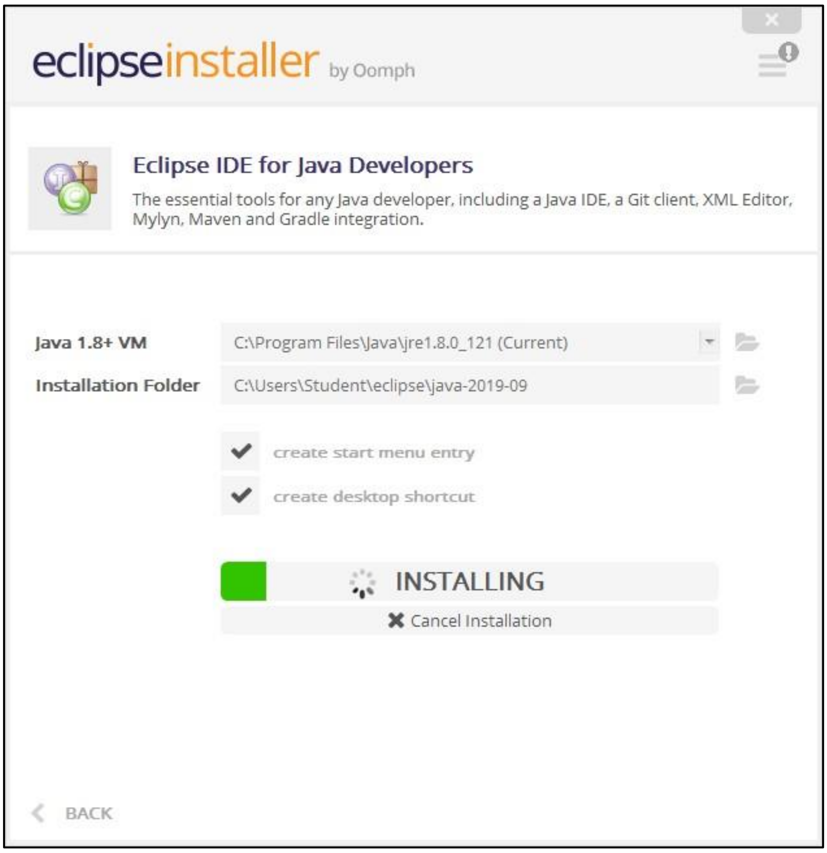
THE ECLIPSE FOUNDATION MAKES AVAILABLE SOFTWARE, DOCUMENTATION, INFORMATION AND/OR OTHER MATERIALS FOR OPEN SOURCE PROJECTS (COLLECTIVELY "CONTENT"). USE OF THE CONTENT IS GOVERNED BY THE TERMS AND CONDITIONS OF THIS AGREEMENT AND/OR THE TERMS AND CONDITIONS OF LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW. BY USING THE CONTENT, YOU AGREE THAT YOUR USE OF THE CONTENT IS GOVERNED BY THIS AGREEMENT AND/OR THE TERMS AND CONDITIONS OF ANY APPLICABLE LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW. IF YOU DO NOT AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT AND THE TERMS AND CONDITIONS OF ANY APPLICABLE LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW, THEN YOU MAY NOT USE THE CONTENT.

Applicable Licenses

Unless otherwise indicated, all Content made available by the Eclipse Foundation is provided to you under the terms and conditions of the Eclipse Public License Version 1.0 ("EPL"). A copy of the EPL is provided with this Content and is also available at http://www.eclipse.org/legal/epl-v20.html. For purposes of the EPL, "Program" will mean the Content.

Content includes, but is not limited to, source code, object code, documentation and other files maintained in the Eclipse Foundation source code repository ("Repository") in software modules ("Modules") and made

Then the Eclipse installation begins.



Eclipse IDE for Java Developers

The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven and Gradle integration.

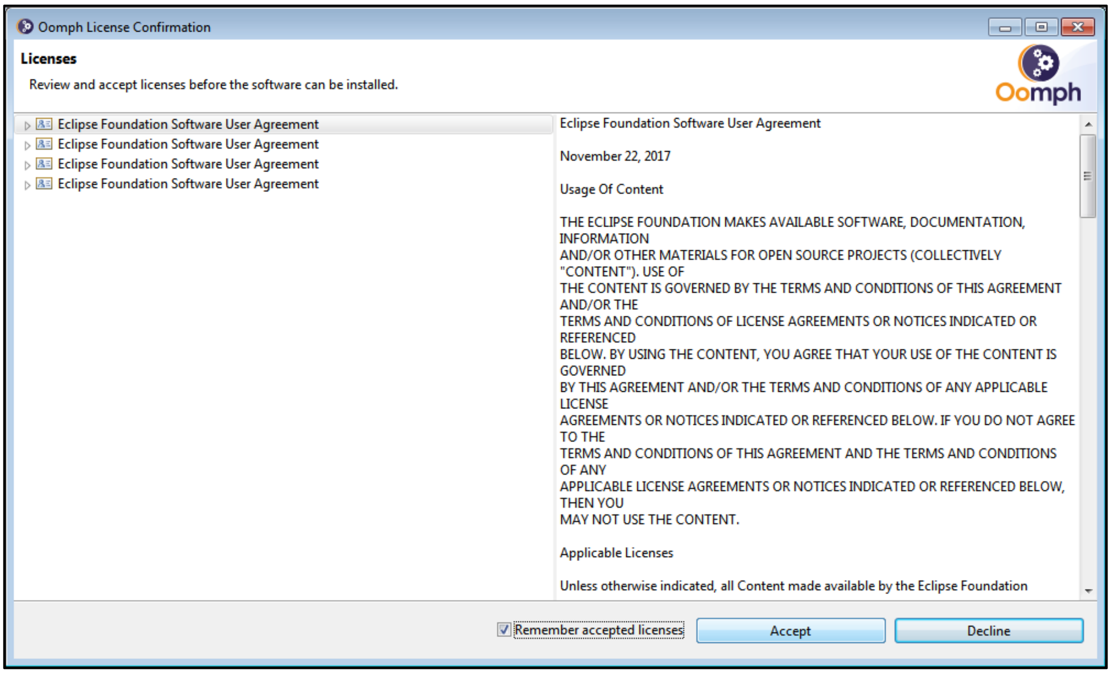
Java 1.8+ VM: C:\Program Files\Java\jre1.8.0_121 (Current)

Installation Folder: C:\Users\Student\eclipse\java-2019-09

✓ create start menu entry

✓ create desktop shortcut

Click on Accept



Licenses

Review and accept licenses before the software can be installed.

- Eclipse Foundation Software User Agreement
- Eclipse Foundation Software User Agreement
- Eclipse Foundation Software User Agreement
- Eclipse Foundation Software User Agreement

Eclipse Foundation Software User Agreement

November 22, 2017

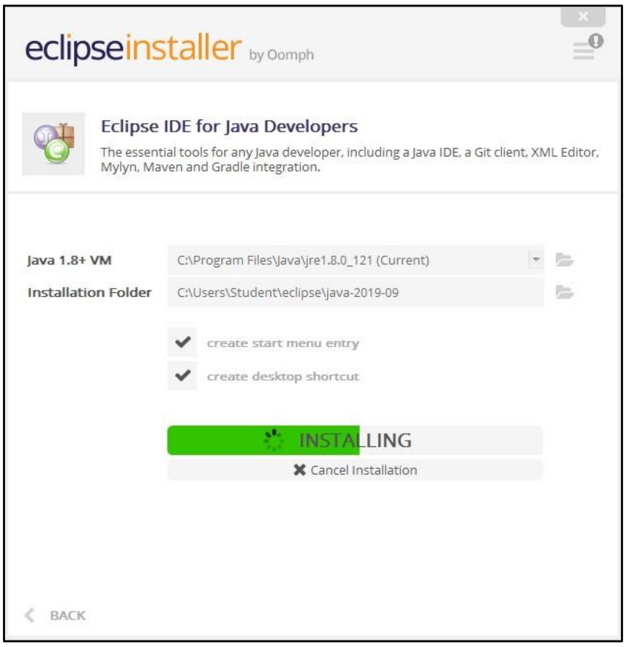
Usage Of Content

THE ECLIPSE FOUNDATION MAKES AVAILABLE SOFTWARE, DOCUMENTATION,
INFORMATION
AND/OR OTHER MATERIALS FOR OPEN SOURCE PROJECTS (COLLECTIVELY
"CONTENT"). USE OF
THE CONTENT IS GOVERNED BY THE TERMS AND CONDITIONS OF THIS AGREEMENT
AND/OR THE
TERMS AND CONDITIONS OF LICENSE AGREEMENTS OR NOTICES INDICATED OR
REFERENCED
BELOW. BY USING THE CONTENT, YOU AGREE THAT YOUR USE OF THE CONTENT IS
GOVERNED
BY THIS AGREEMENT AND/OR THE TERMS AND CONDITIONS OF ANY APPLICABLE
LICENSE
AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW. IF YOU DO NOT AGREE
TO THE
TERMS AND CONDITIONS OF THIS AGREEMENT AND THE TERMS AND CONDITIONS
OF ANY
APPLICABLE LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW,
THEN YOU
MAY NOT USE THE CONTENT.

Applicable Licenses

Unless otherwise indicated, all Content made available by the Eclipse Foundation

☑ Remember accepted licenses



Eclipse IDE for Java Developers

The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven and Gradle integration.

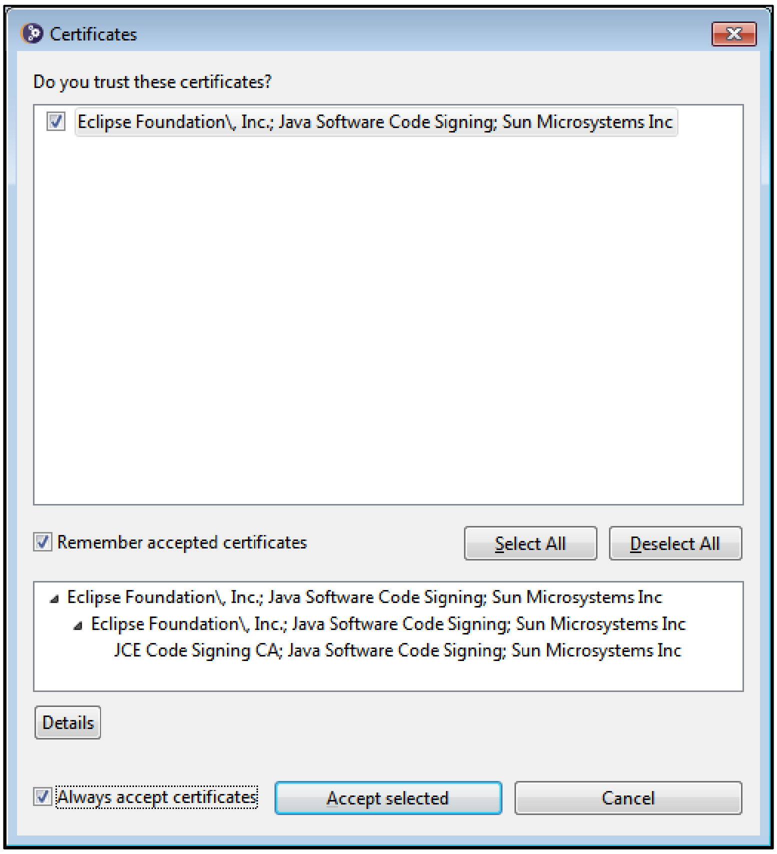
Java 1.8+ VM: C:\Program Files\Java\jre1.8.0_121 (Current)

Installation Folder: C:\Users\Student\eclipse\java-2019-09

✓ create start menu entry

✓ create desktop shortcut

Click on Select All and Accept Selected.



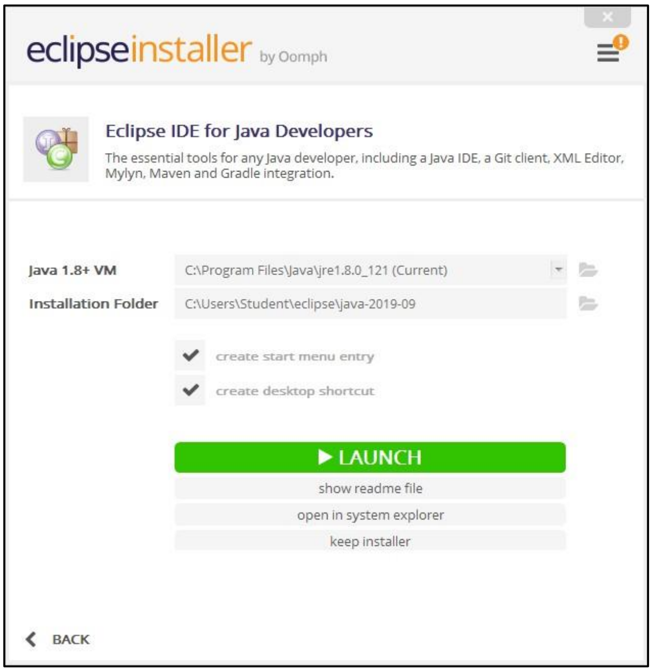
Do you trust these certificates?

☑ Eclipse Foundation\, Inc.; Java Software Code Signing; Sun Microsystems Inc

☑ Remember accepted certificates

- Eclipse Foundation\, Inc.; Java Software Code Signing; Sun Microsystems Inc
  - Eclipse Foundation\, Inc.; Java Software Code Signing; Sun Microsystems Inc
    - JCE Code Signing CA; Java Software Code Signing; Sun Microsystems Inc

☑ Always accept certificates



Eclipse IDE for Java Developers

The essential tools for any Java developer, including a Java IDE, a Git client, XML Editor, Mylyn, Maven and Gradle integration.

Java 1.8+ VM: C:\Program Files\Java\jre1.8.0_121 (Current)

Installation Folder: C:\Users\Student\eclipse\java-2019-09

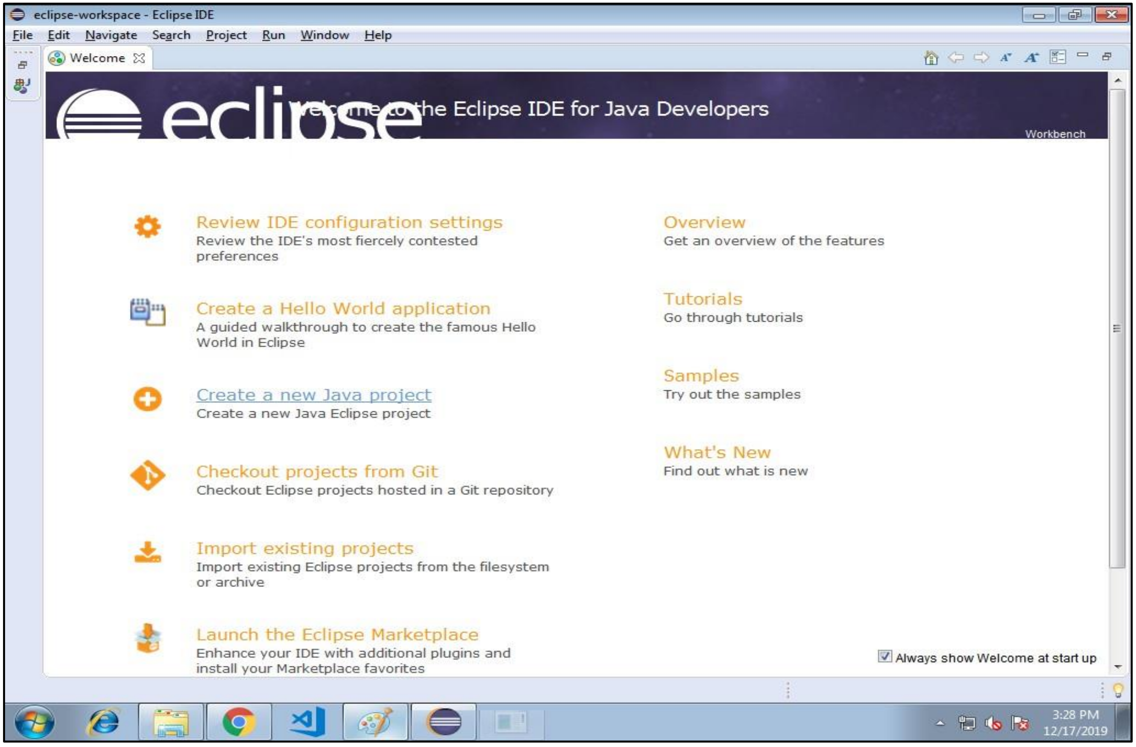
☑ create start menu entry

☑ create desktop shortcut

CREATING PROJECT AND CLASSES IN ECLIPSE IDE

Browse the Workspace for storing the java project and click on Launch.

Select "Create a new Java project".



Welcome to the Eclipse IDE for Java Developers

Review IDE configuration settings
Review the IDE's most fiercely contested preferences

Create a Hello World application
A guided walkthrough to create the famous Hello World in Eclipse

Create a new Java project
Create a new Java Eclipse project

Checkout projects from Git
Checkout Eclipse projects hosted in a Git repository

Import existing projects
Import existing Eclipse projects from the filesystem or archive

Launch the Eclipse Marketplace
Enhance your IDE with additional plugins and install your Marketplace favorites

Overview
Get an overview of the features

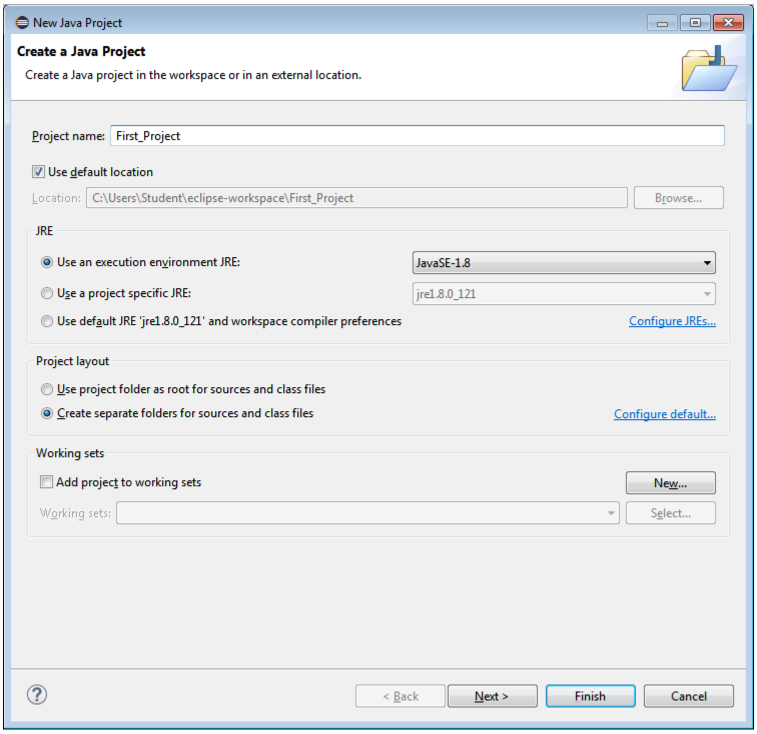
Tutorials
Go through tutorials

Samples
Try out the samples

What's New
Find out what is new

☑ Always show Welcome at start up

Type the project name and click on Finish.



Create a Java Project

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

Project name: First_Project

☑ Use default location

Location: C:\Users\Student\eclipse-workspace\First_Project

JRE

● Use an execution environment JRE: JavaSE-1.8

○ Use a project specific JRE: jre1.8.0_121

○ Use default JRE 'jre1.8.0_121' and workspace compiler preferences

Project layout

○ Use project folder as root for sources and class files

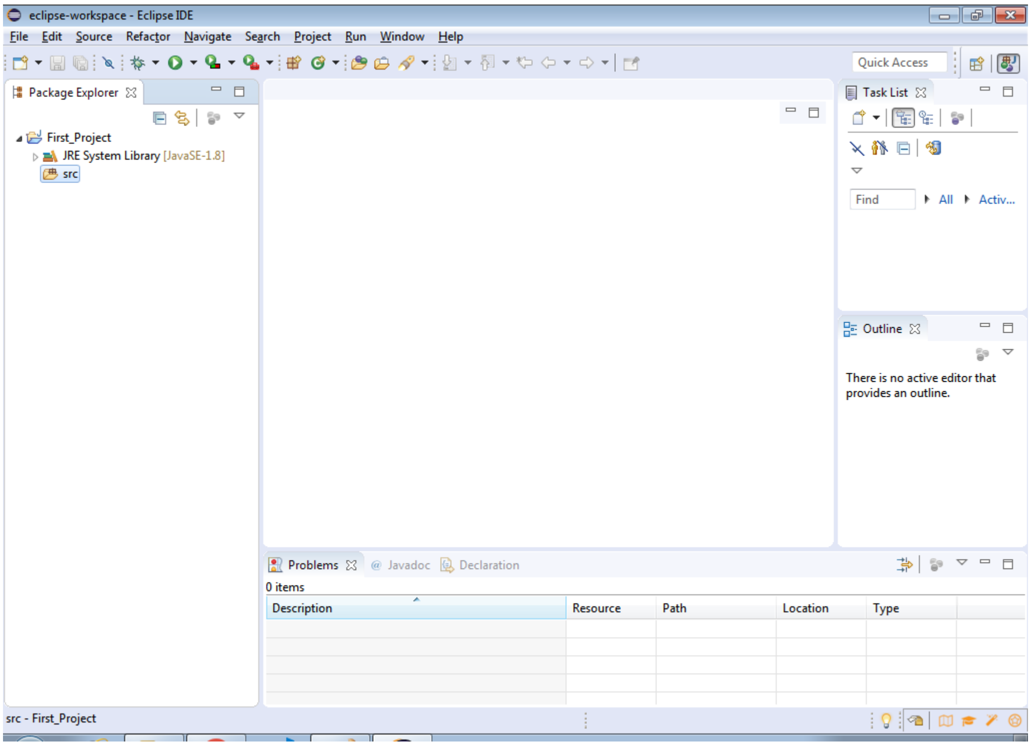
● Create separate folders for sources and class files

Working sets

□ Add project to working sets

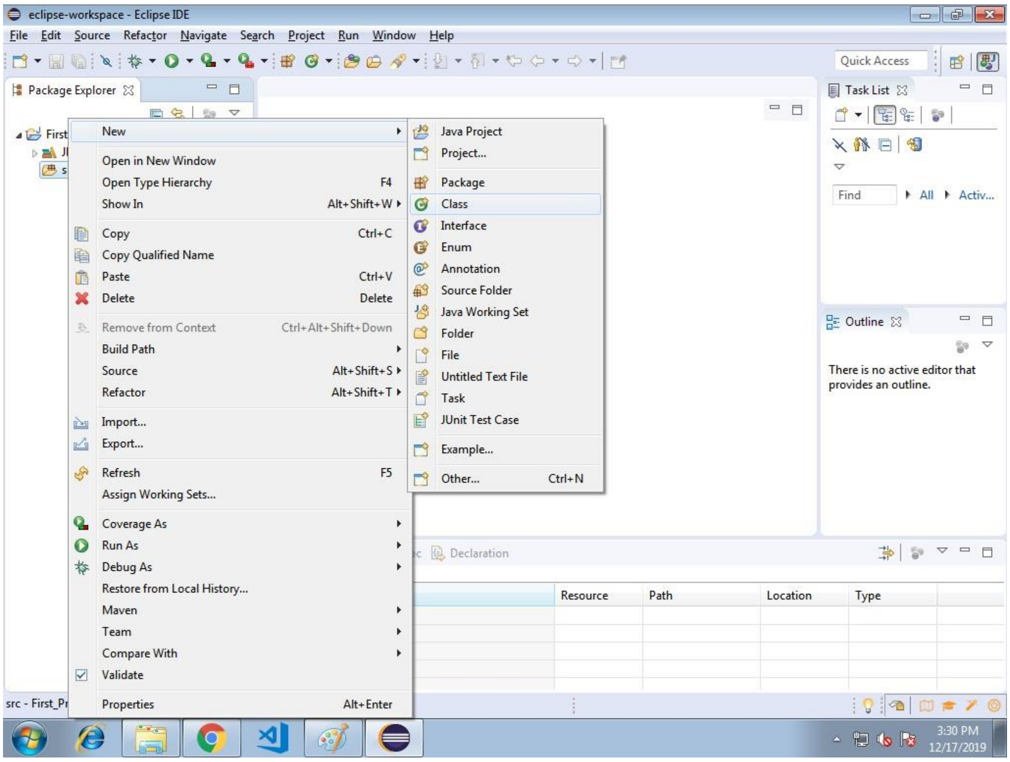
Working sets:

Now, create the class in src directory from Package Explorer window.



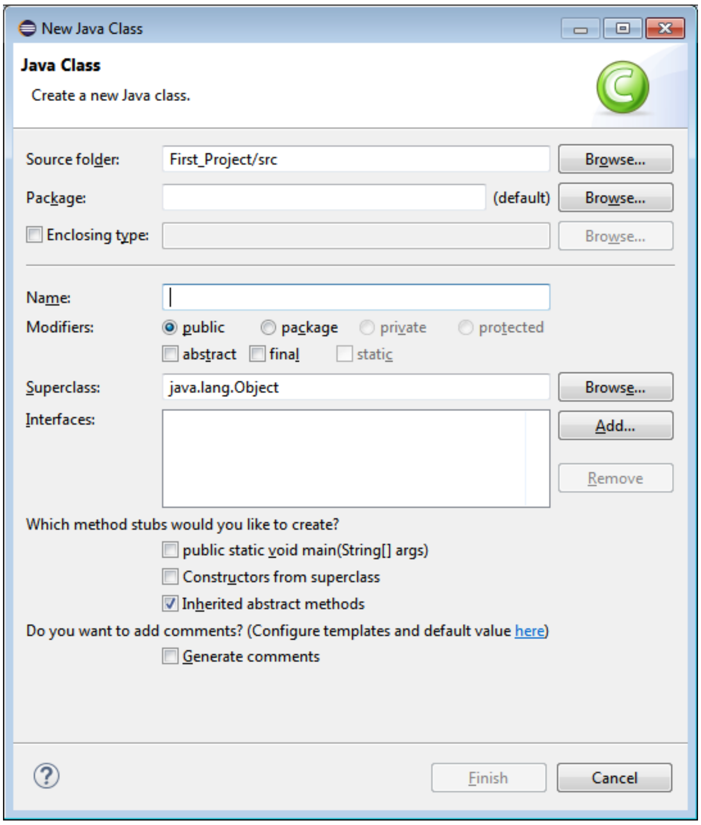
There is no active editor that provides an outline.

0 items



There is no active editor that provides an outline.

Type the class name and click on Finish.



New Java Class

Java Class

Create a new Java class.

Source folder: First_Project/src

Package: (default)

□ Enclosing type:

Name:

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)

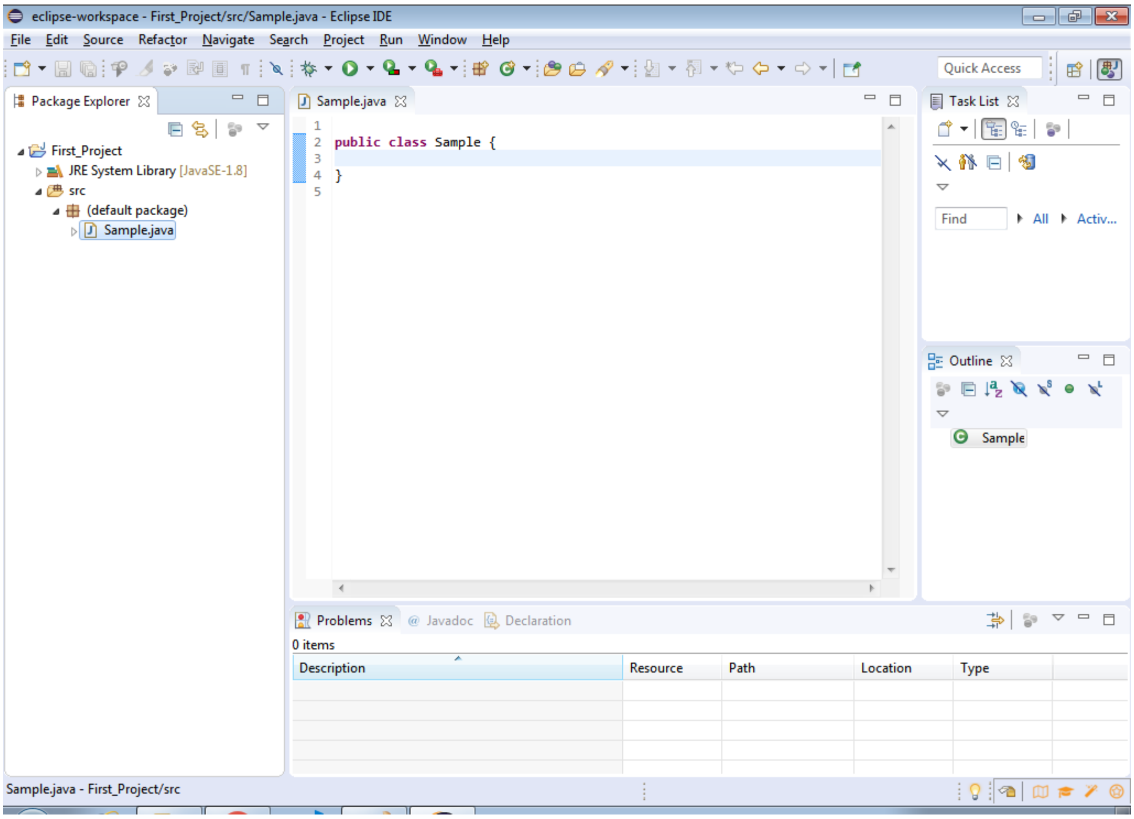
□ Constructors from superclass

☑ Inherited abstract methods

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

□ Generate comments

Type the java code.



eclipse-workspace - First_Project/src/Sample.java - Eclipse IDE

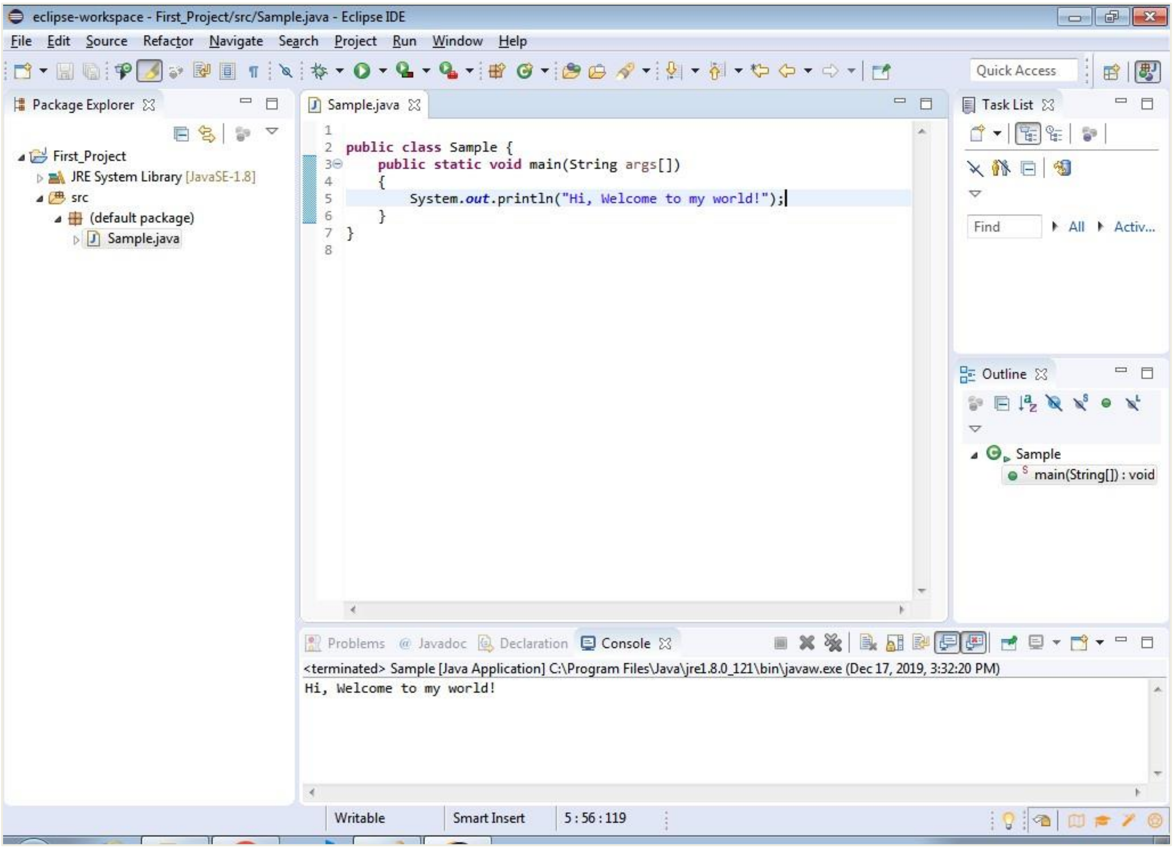
```
public class Sample {

}
```

0 items

Description	Resource	Path	Location	Type

Sample.java - First_Project/src

Click on Play button to run or execute the java code.



```java

public class Sample {
    public static void main(String args[])
    {
        System.out.println("Hi, Welcome to my world!");
    }
}

```

```
<terminated> Sample [Java Application] C:\Program Files\Java\jre1.8.0_121\bin\javaw.exe (Dec 17, 2019, 3:32:20 PM)
Hi, Welcome to my world!
```

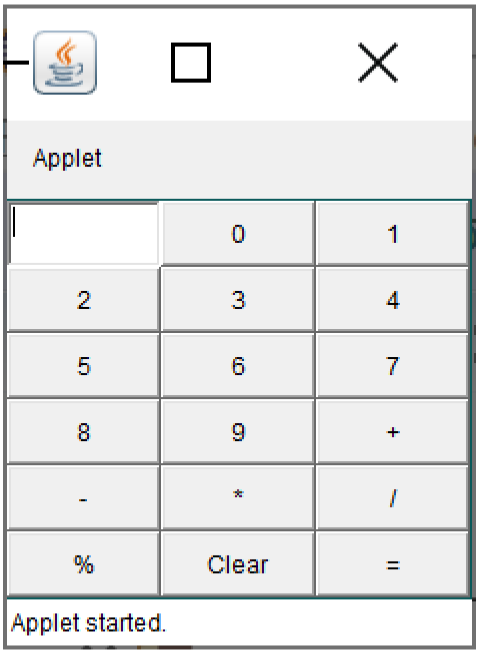
Week 2:
Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -,*, % operations. Add a text field to display the result. Handle any possible exceptions like divided by zero.

Source Code:

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;

     /*
     * <applet code="Calculator" width=500 height=500></applet>
     * */
public class Calculator extends Applet implements ActionListener
    {
String msg=" ";
int v1,v2,result;
TextField t1;
     Button b[]=new Button[10];
     Button add,sub,mul,div,clear,mod,EQ;
     char OP;
     public void init()
     {
     Color k=new Color(10,89,90);
     setBackground(k);
     t1=new TextField(50);
     GridLayout gl=new GridLayout(6,3);
     setLayout(gl);
     for(int i=0;i<10;i++)
     {
     b[i]=new Button(""+i);
      }
      add=new Button("+");
      sub=new Button("-");
      mul=new Button("*");
      div=new Button("/");
      mod=new Button("%");
      clear=new Button("Clear");
      EQ=new Button("=");
      t1.addActionListener(this);add(t1);
      for(int i=0;i<10;i++)
          {
            add(b[i]);
          }
            add(add);
            add(sub);
            add(mul);
            add(div);
            add(mod);
            add(clear);
```

```
                add(EQ);
                for(int i=0;i<10;i++)
                {
                b[i].addActionListener(this);
                }
                add.addActionListener(this);
                sub.addActionListener(this);
                mul.addActionListener(this);
                div.addActionListener(this);
                mod.addActionListener(this);
                clear.addActionListener(this);
                EQ.addActionListener(this);
}
public void actionPerformed(ActionEvent ae)
{
String str=ae.getActionCommand();
char ch=str.charAt(0);

if ( Character.isDigit(ch))
t1.setText(t1.getText()+str);
else
if(str.equals("+"))
{
v1=Integer.parseInt(t1.getText());OP='+';
t1.setText("");
}
else if(str.equals("-"))
{
v1=Integer.parseInt(t1.getText());
OP='-';t1.setText("");
}
else if(str.equals("*"))
{
v1=Integer.parseInt(t1.getText());OP='*';
t1.setText("");
}
else if(str.equals("/"))
{
v1=Integer.parseInt(t1.getText());OP='/';
t1.setText("");
}
else if(str.equals("%"))
{
 v1=Integer.parseInt(t1.getText());OP='%';
t1.setText("");
}
if(str.equals("="))
{
v2=Integer.parseInt(t1.getText());
if(OP=='+')
result=v1+v2;
else if(OP=='-')
result=v1-v2;
else if(OP=='*')
result=v1*v2;
else if(OP=='/')
result=v1/v2;
else if(OP=='%')
          result=v1%v2;
          t1.setText(""+result);
}
if(str.equals("Clear"))
{
t1.setText("");
}
}
}
```

Output:



Applet started.

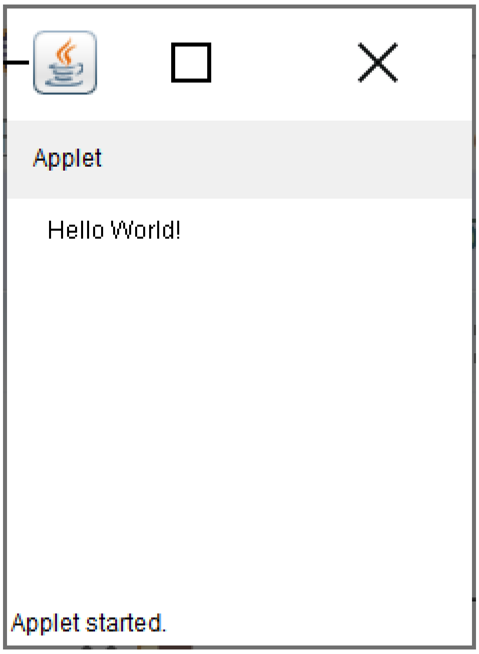
Week 3:

1. Develop an applet in Java that displays a simple message.
2. Develop an applet in Java that receives an integer in one text field, and computes its factorial Value andreturns it in another text field, when the button named “Compute” is clicked.

Source Code: Develop an applet in Java that displays a simple message

```
// Import the packages to access the classes and methods in awt and applet classes.
import java.awt.*;
import java.applet.*;

/* <applet code="Applet1" width=200 height=300></applet>*/

public class AppletExample extends Applet
{
// Paint method to display the message.
public void paint(Graphics g)
{
g.drawString("Hello World!",20,20);
}
}
```

Output:



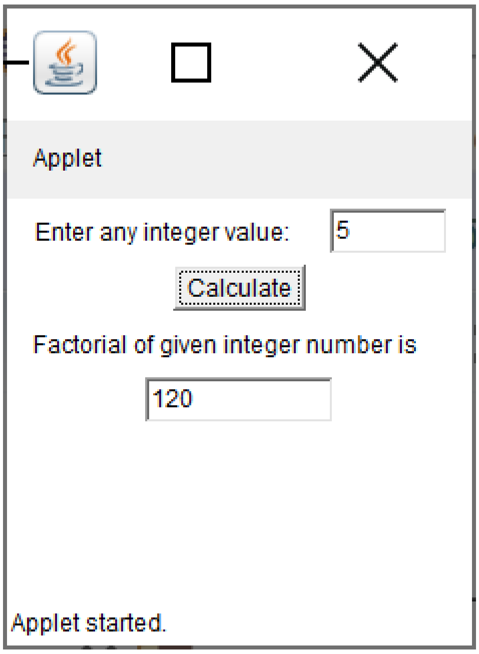
Hello World!

Applet started.

Source Code: Develop an applet in Java that receives an integer in one text field, and computes its factorial Value andreturns it in another text field, when the button named “Compute” is clicked

```
import java.awt.*;
import java.awt.event.*;
import java.applet.Applet;
/*<applet code="Fact.class" height=300 width=300></applet>*/
public class Factorial extends Applet implements
ActionListener{Label l1,l2;
TextField t1,t2;
Button b1;
public void init(){
l1=new Label("Enter any integer value: ");add(l1);
t1=new TextField(5);add(t1);
b1=new Button("Calculate");add(b1);
b1.addActionListener(this);
l2=new Label("Factorial of given integer number is ");add(l2);
t2=new TextField(10);add(t2);
}
public void actionPerformed(ActionEvent e){
if(e.getSource()==b1){
int fact=fact(Integer.parseInt(t1.getText()));
t2.setText(String.valueOf(fact));
}      }
int fact(int f){
int s=0; if(f==0)
return 1;
else
}
  }
return f*fact(f-1);
```

Output:



Enter any integer value: 5

Factorial of given integer number is

120

Applet started.

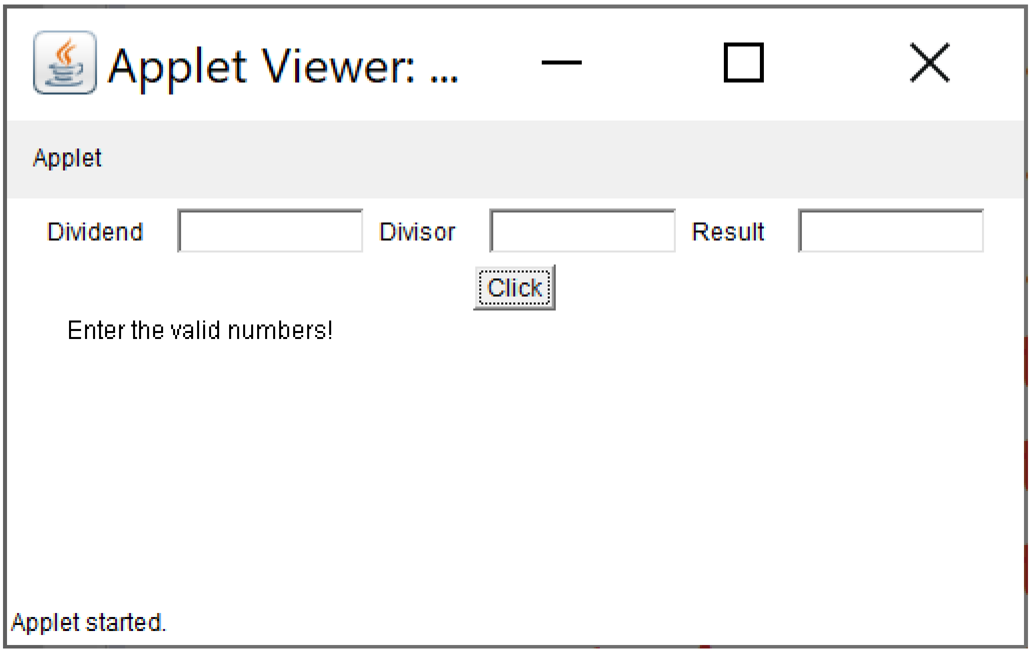
Week 4:

Write a Java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num 2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception. Display the exception in a message dialog box.

Source Code:

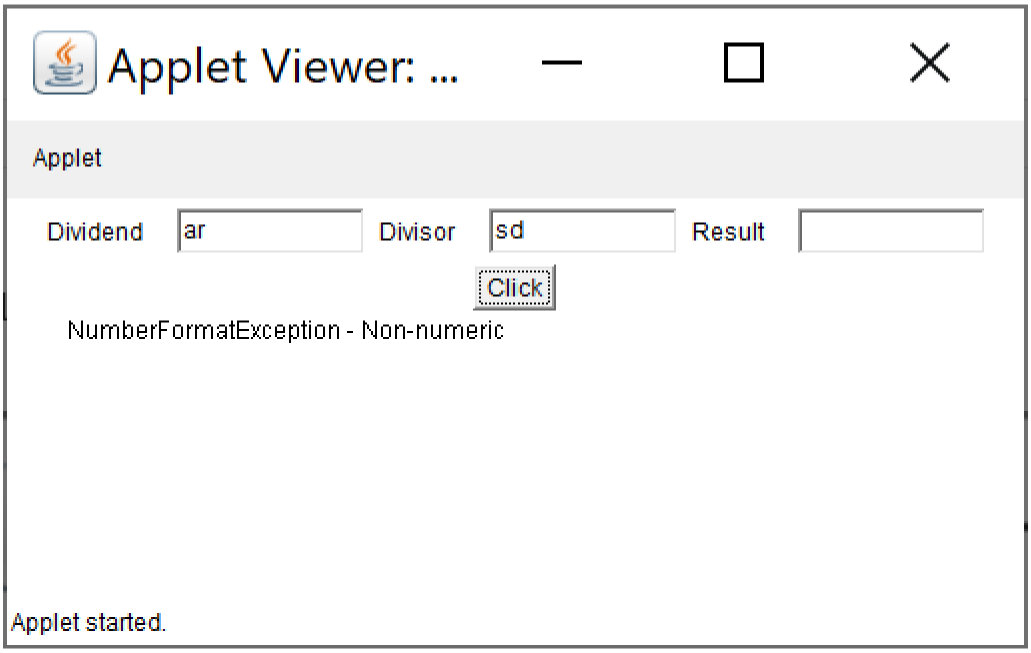
```
import java.awt.*; import
java.awt.event.*;import
java.applet.*;

/*<applet code="DivisionExample"width=230 height=250></applet>*/

public class DivisionExample extends Applet implements ActionListener
{
String msg;
TextField num1, num2, res;
Label l1, l2, l3;
Button div;

public void init()
{
l1 = new Label("Dividend");
l2 = new Label("Divisor");
l3 = new Label("Result");
num1 = new TextField(10);
num2 = new TextField(10);
res = new TextField(10);
div = new Button("Click");
div.addActionListener(this);add(l1);
add(num1);add(l2);
add(num2);add(l3);
add(res);
add(div);
}

public void actionPerformed(ActionEvent ae)
{
String arg = ae.getActionCommand();
int num1 = 0, num2 = 0;
if (arg.equals("Click")) {
if (this.num1.getText().isEmpty() | this.num2.getText().isEmpty())
{
msg = "Enter the valid numbers!";
repaint();
}
else
{
try
{
num1 = Integer.parseInt(this.num1.getText());
num2 = Integer.parseInt(this.num2.getText());
int num3 = num1 / num2;
res.setText(String.valueOf(num3));
msg = "Operation Succesfull!!!";
repaint();
}
catch (NumberFormatException ex)
{
System.out.println(ex);
res.setText("");
msg = "NumberFormatException-Non-numeric";
repaint();
}
 catch (ArithmeticException e)
{
System.out.println("Can't be divided by Zero" + e);
res.setText("");
msg = "Can't be divided by Zero";
repaint();
}
}
}
}
public void paint(Graphics g)
{
g.drawString(msg, 30, 70);
                              }
                      }
```

Output:



Enter the valid numbers!

Applet started.

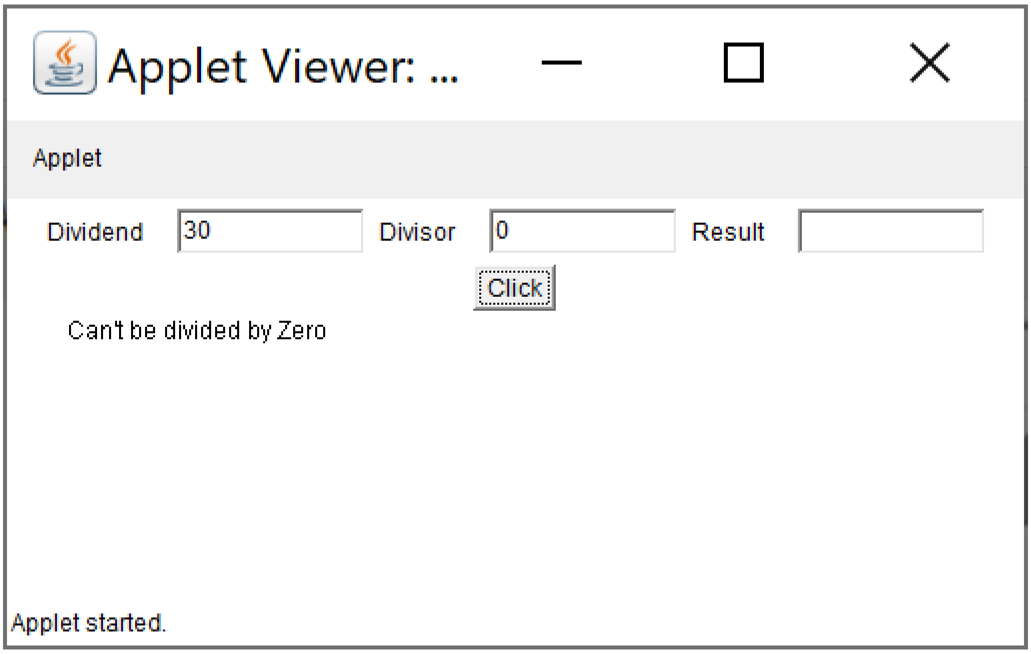


Dividend: ar

Divisor: sd

NumberFormatException - Non-numeric

Applet started.

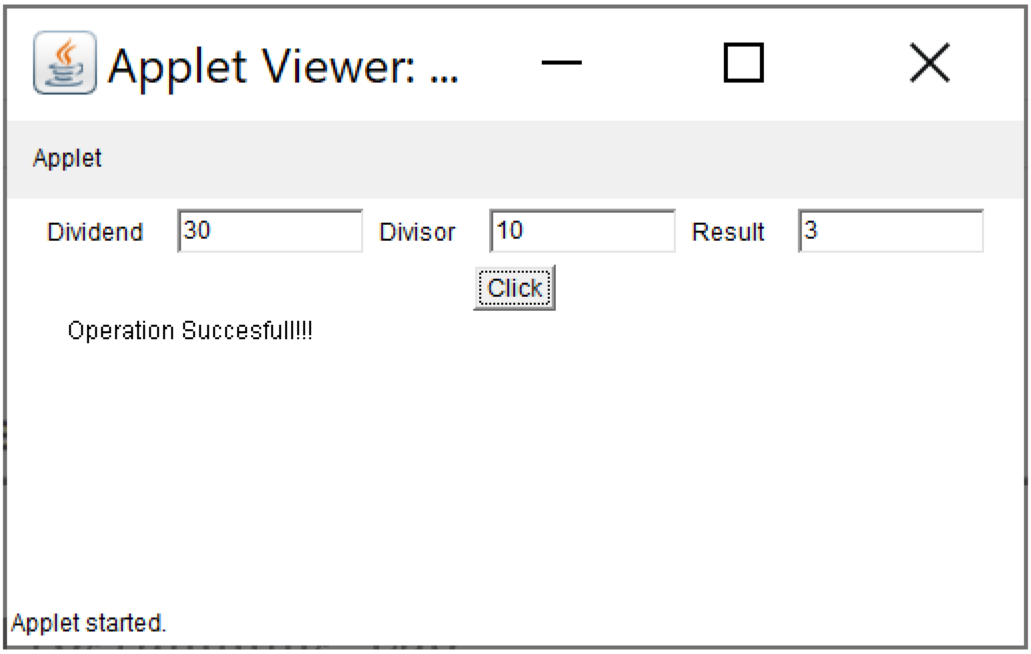


30

0

Can't be divided by Zero

Applet started.



30

10

3

Operation Succesfull!!!

Applet started.

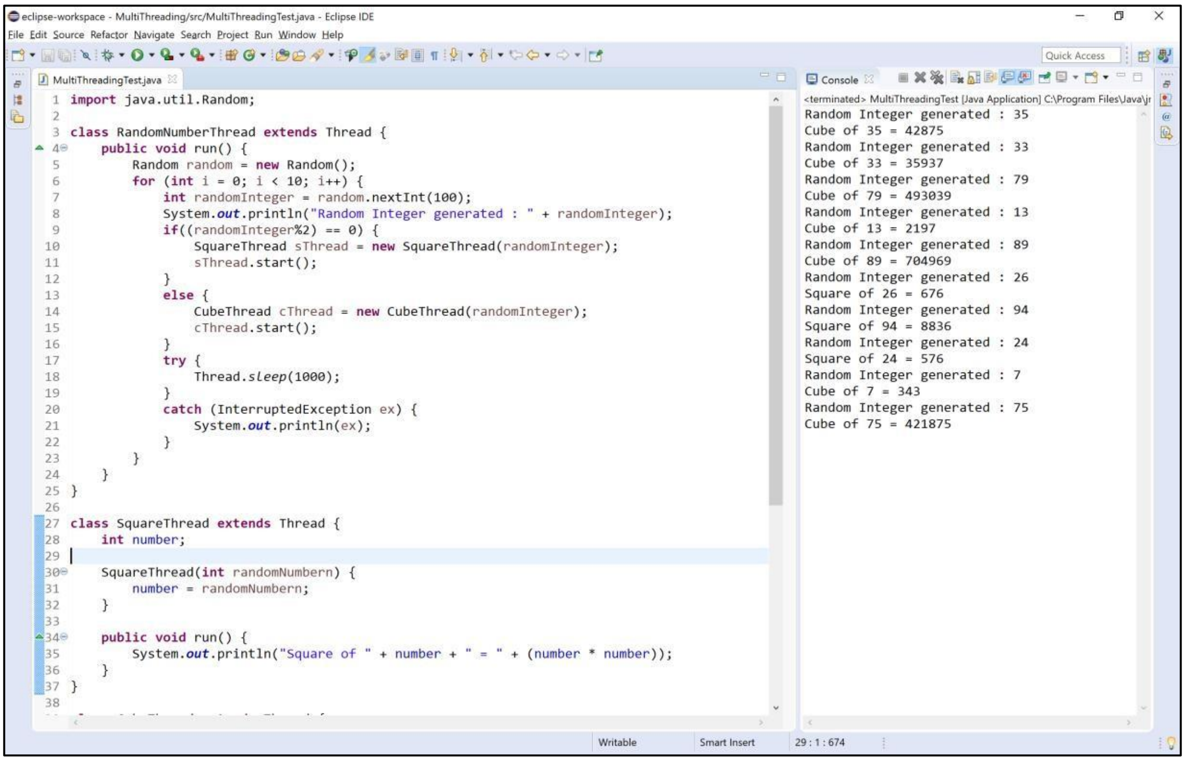
Week 5:

Write a Java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.

Source Code:

```
import java.util.Random;

class RandomNumberThread extends Thread
 {
public void run()
{
Random random = new Random();
for (int i = 0; i < 10; i++)
{
int randomInteger = random.nextInt(100);
System.out.println("Random Integer generated : " + randomInteger);
if((randomInteger%2) == 0)
{
SquareThread sThread = new SquareThread(randomInteger);
sThread.start();
}
else {

}
try {

}
CubeThread cThread = new CubeThread(randomInteger);cThread.start();
 Thread.sleep(1000);
catch (InterruptedException ex) {System.out.println(ex);
                                                }
                                        }
                                }
                        }
class SquareThread extends Thread {
int number;

SquareThread(int randomNumbern) {number = randomNumbern;
}
public void run()
{
System.out.println("Square of " + number + " = " + (number * number));
}
}
class CubeThread extends Thread
{
int number;

CubeThread(int randomNumber)
{
number = randomNumber;
}
public void run()
{
System.out.println("Cube of " + number + " = " + number * number *number);
}
}
public class MultiThreadingTest
{
public static void main(String args[])
{
RandomNumberThread rnThread = new RandomNumberThread();
rnThread.start();
}
}
```

Output:



eclipse-workspace - MultiThreading/src/MultiThreadingTest.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
MultiThreadingTest.java
Console
Writable
Smart Insert
29 : 1 : 674

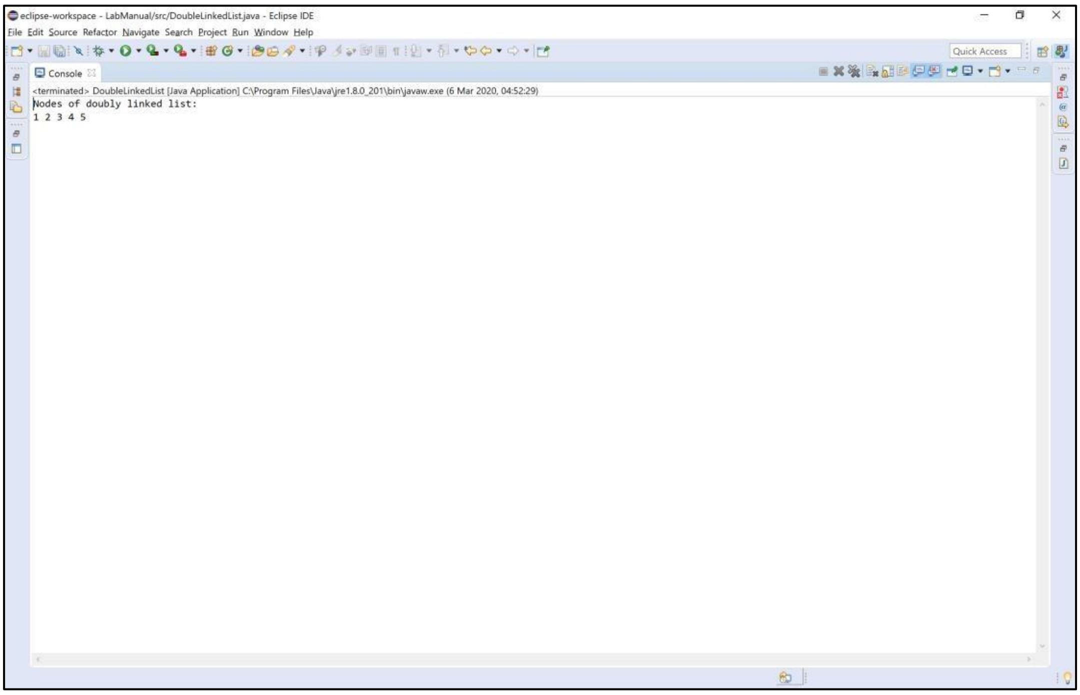
```java
import java.util.Random;

class RandomNumberThread extends Thread {
    public void run() {
        Random random = new Random();
        for (int i = 0; i < 10; i++) {
            int randomInteger = random.nextInt(100);
            System.out.println("Random Integer generated : " + randomInteger);
            if((randomInteger%2) == 0) {
                SquareThread sThread = new SquareThread(randomInteger);
                sThread.start();
            }
            else {
                CubeThread cThread = new CubeThread(randomInteger);
                cThread.start();
            }
            try {
                Thread.sleep(1000);
            }
            catch (InterruptedException ex) {
                System.out.println(ex);
            }
        }
    }
}

class SquareThread extends Thread {
    int number;

    SquareThread(int randomNumbern) {
        number = randomNumbern;
    }

    public void run() {
        System.out.println("Square of " + number + " = " + (number * number));
    }
}

```

```
<terminated> MultiThreadingTest [Java Application] C:\Program Files\Java\jr
Random Integer generated : 35
Cube of 35 = 42875
Random Integer generated : 33
Cube of 33 = 35937
Random Integer generated : 79
Cube of 79 = 493039
Random Integer generated : 13
Cube of 13 = 2197
Random Integer generated : 89
Cube of 89 = 704969
Random Integer generated : 26
Square of 26 = 676
Random Integer generated : 94
Square of 94 = 8836
Random Integer generated : 24
Square of 24 = 576
Random Integer generated : 7
Cube of 7 = 343
Random Integer generated : 75
Cube of 75 = 421875
```

Week 6: Write a C++ to illustrate the concepts of console I/O operations.
Source code:

```
public class DoubleLinkedList {
class Node {
int   data; Node previous;Node next;
public Node(int data) {
this.data = data;
}  }
Node head, tail = null;
public void addNode(int data) {
Node newNode = new Node(data);
if (head == null) {
head = tail = newNode;head.previous = null; tail.next = null;
} else {
tail.next = newNode; newNode.previous = tail;tail = newNode; tail.next = null;
}
}
public void display() { Node current = head;if (head == null) {
System.out.println("List is empty");
return;
}
System.out.println("Nodes of doubly linked list: ");
while (current != null) {
System.out.print(current.data + " ");current = current.next;
}
}
public static void main(String[] args) {

DoubleLinkedList dList = new DoubleLinkedList();dList.addNode(1);
dList.addNode(2);dList.addNode(3);dList.addNode(4);dList.addNode(5);
dList.display();
}
}
```

Output:



eclipse-workspace - LabManual/src/DoubleLinkedList.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Console

```
<terminated> DoubleLinkedList [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (6 Mar 2020, 04:52:29)
Nodes of doubly linked list:
1 2 3 4 5
```

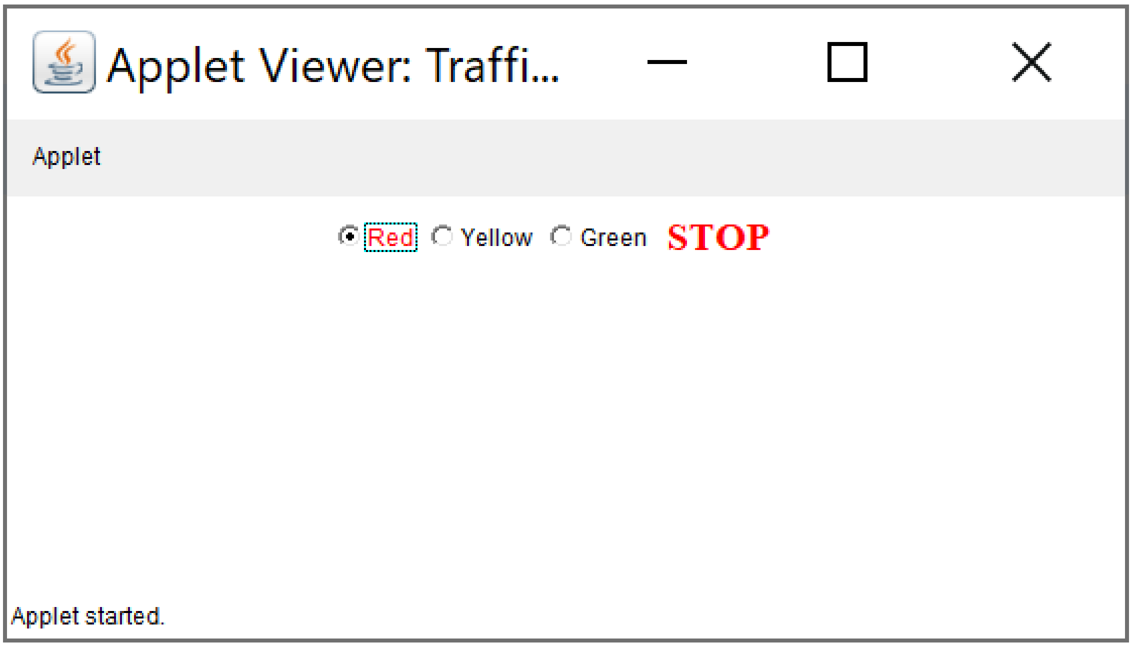
Week 7:

Write a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green with radio buttons. On selecting a button, an appropriate message with “Stop” or “Ready” or “Go” should appear above the buttons in selected color. Initially, there is no message shown.

Source Code:

```
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;

 /*
* <applet code = "TrafficLightsExample" width = 1000 height = 500>
* </applet>
 * */
public class TrafficLightsExample extends Applet implements ItemListener
{
CheckboxGroup grp = new CheckboxGroup();
Checkbox redLight, yellowLight, greenLight;Label msg;
public void init(){
redLight = new Checkbox("Red", grp, false);
yellowLight = new Checkbox("Yellow", grp, false);
greenLight = new Checkbox("Green", grp, false); msg = new Label("");

redLight.addItemListener(this);
yellowLight.addItemListener(this);
greenLight.addItemListener(this);
add(redLight); add(yellowLight);
add(greenLight); add(msg);
msg.setFont(new Font("Serif", Font.BOLD, 20));
}
public void itemStateChanged(ItemEvent ie)
{
redLight.setForeground(Color.BLACK);
yellowLight.setForeground(Color.BLACK);
greenLight.setForeground(Color.BLACK);

if(redLight.getState() == true)
{
redLight.setForeground(Color.RED);
msg.setForeground(Color.RED);
msg.setText("STOP");
}
else if(yellowLight.getState() == true)
{
yellowLight.setForeground(Color.YELLOW);
msg.setForeground(Color.YELLOW);
msg.setText("READY");
}
Else
{

}
}
}
  greenLight.setForeground(Color.GREEN);
  msg.setForeground(Color.GREEN);
  msg.setText("GO");
```

Output:



STOP

Applet started.

Week 8:

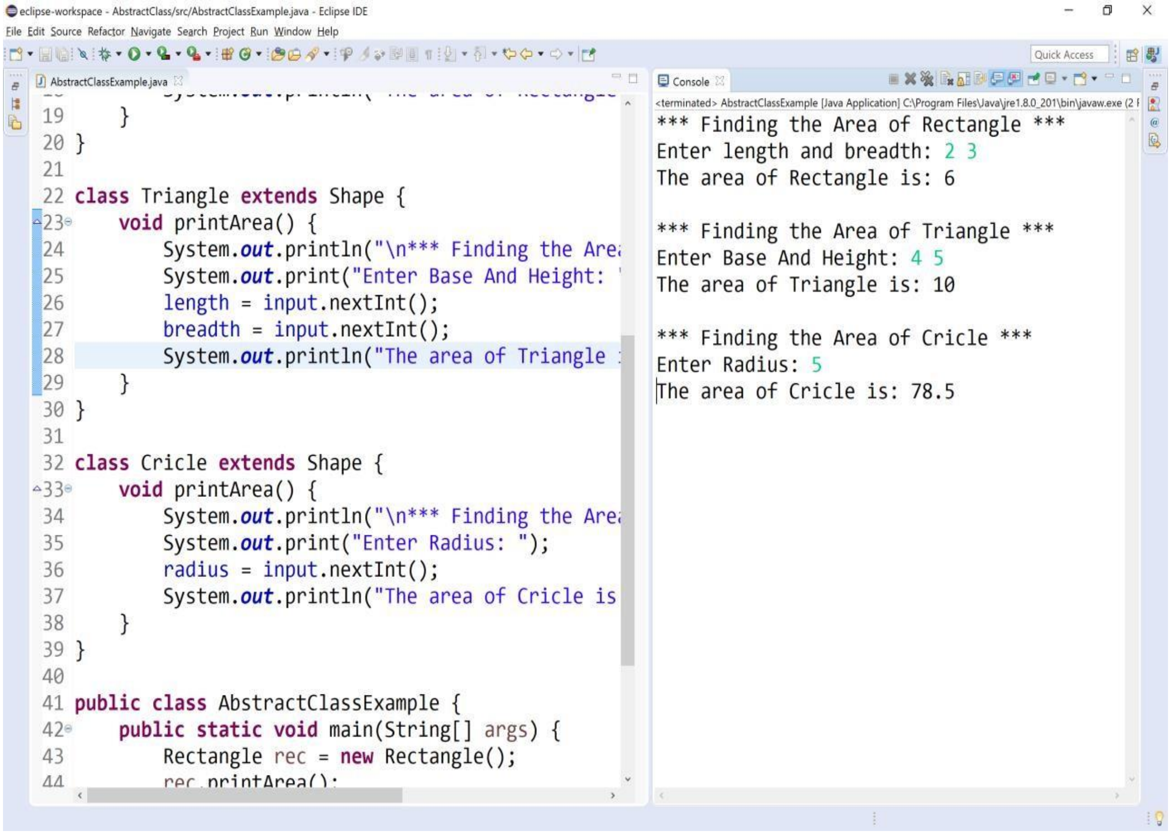
Write a Java program to create an abstract class named Shape that contains two integers and an empty method named print Area (). Provide three classes named Rectangle, Triangle, and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.

Source Code:

```
import java.util.*;

abstract class Shape
{
int length, breadth, radius;
Scanner input = new Scanner(System.in);
abstract void printArea();
}
class Rectangle extends Shape
 {
void printArea()
{
System.out.println("*** Finding the Area of Rectangle ***");
System.out.print("Enter length and breadth: ");
length = input.nextInt(); breadth = input.nextInt();
System.out.println("The area of Rectangle is: " + length * breadth);
}
}
class Triangle extends Shape
 {
void printArea() {
System.out.println("\n*** Finding the Area of Triangle ***");
System.out.print("Enter Base And Height: ");
length = input.nextInt(); breadth = input.nextInt();
System.out.println("The area of Triangle is: " + (length * breadth)/2);
}
}
class Cricle extends Shape
{
void printArea() {
System.out.println("\n*** Finding the Area of Cricle ***");
System.out.print("Enter Radius: ");
radius = input.nextInt();
System.out.println("The area of Cricle is: " + 3.14f * radius * radius);
}
}
public class AbstractClassExample
{
public static void main(String[] args)
{

Rectangle rec = new Rectangle(); rec.printArea();
Triangle tri = new Triangle();tri.printArea();
```

Cricle cri = **new** Cricle();cri.printArea();
}
}

Output:


eclipse-workspace - AbstractClass/src/AbstractClassExample.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
AbstractClassExample.java
Console

```
    }
}

class Triangle extends Shape {
    void printArea() {
        System.out.println("\n*** Finding the Area
        System.out.print("Enter Base And Height: '
        length = input.nextInt();
        breadth = input.nextInt();
        System.out.println("The area of Triangle :
    }
}

class Cricle extends Shape {
    void printArea() {
        System.out.println("\n*** Finding the Area
        System.out.print("Enter Radius: ");
        radius = input.nextInt();
        System.out.println("The area of Cricle is
    }
}

public class AbstractClassExample {
    public static void main(String[] args) {
        Rectangle rec = new Rectangle();
        rec.printArea();
```

```
<terminated> AbstractClassExample [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (2 F
*** Finding the Area of Rectangle ***
Enter length and breadth: 2 3
The area of Rectangle is: 6

*** Finding the Area of Triangle ***
Enter Base And Height: 4 5
The area of Triangle is: 10

*** Finding the Area of Cricle ***
Enter Radius: 5
The area of Cricle is: 78.5
```

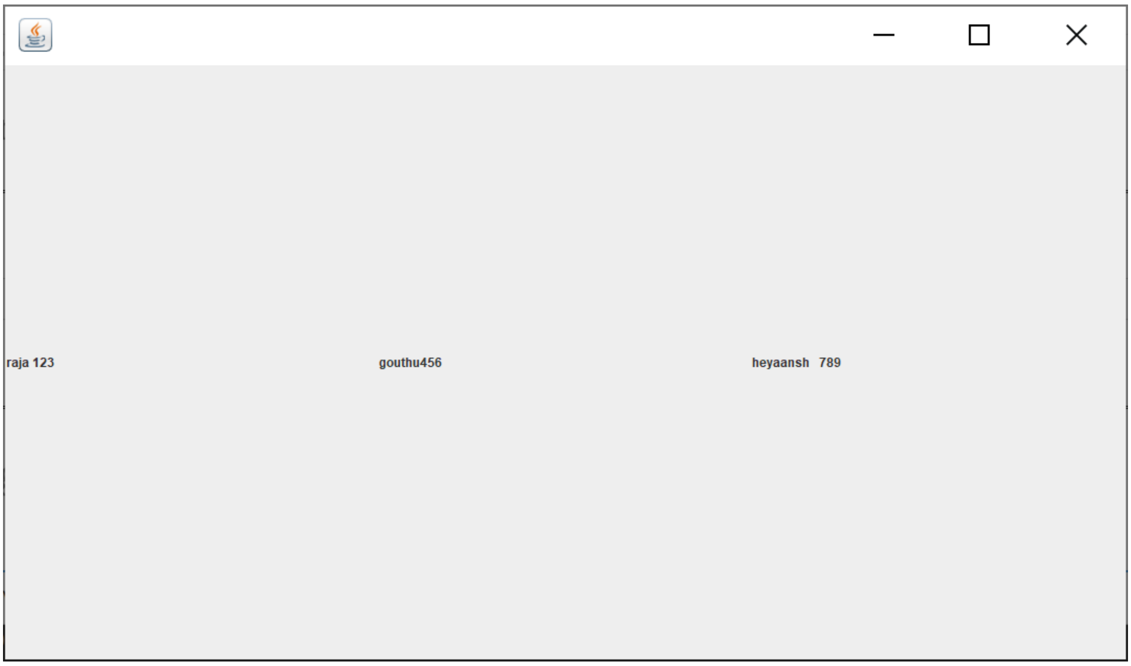
Week 9:

Suppose that a table named Table.txt is stored in a text file. The first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas. Write a java program to display the table using Labels in Grid Layout.

Source Code:

```
import java.io.*;
 import java.util.*;
import java.awt.*;
import javax.swing.*;

class A extends JFrame {
public A() {
setSize(400, 400);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
GridLayout g = new GridLayout(0, 3); setLayout(g);
try
{
FileInputStream fin = new
FileInputStream("C:\\Users\\User\\eclipse-workspace\\LabManual\\src\\HashTab.txt");
Scanner sc = new Scanner(fin).useDelimiter(",");
String[] arrayList;String a;
while (sc.hasNextLine()) {a = sc.nextLine();
arrayList = a.split(",");
for (String i : arrayList) {add(new JLabel(i));
}
}
} catch (Exception ex) {
}
setDefaultLookAndFeelDecorated(true);pack();
setVisible(true);
}
}
public class TableTest
{
public static void main(String[] args) {A a = new A();
}
}
```

Output:



raja 123 gouthu456 heyaansh 789

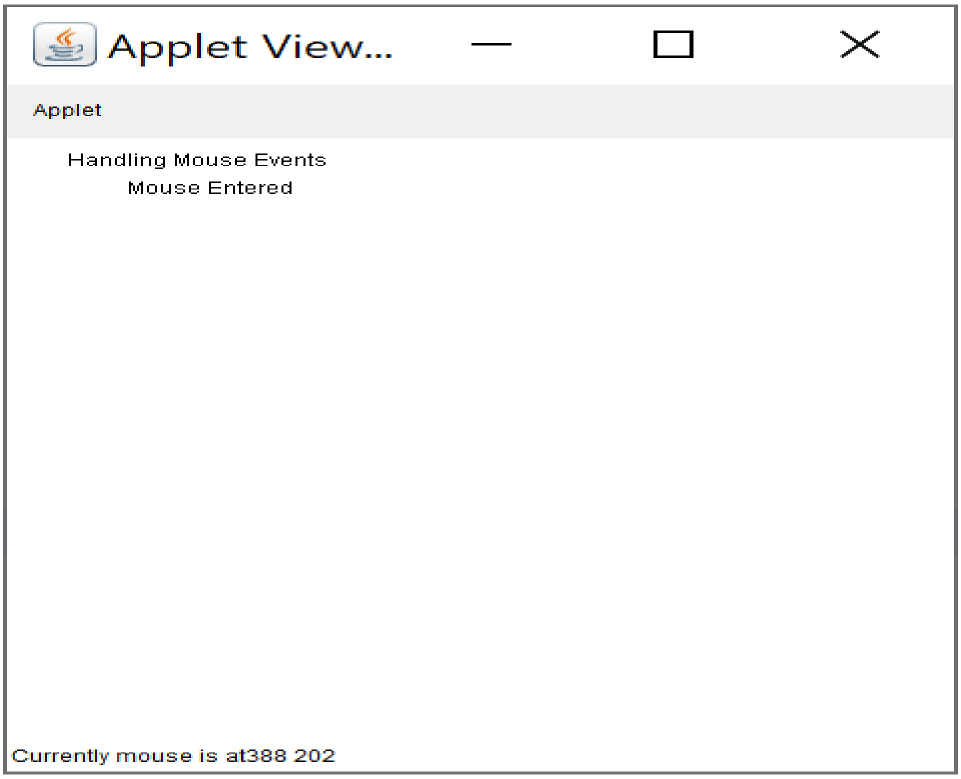
Week 10:

Write a Java program that handles all mouse events and shows the event name at the center of the windowwhen a mouse event is fired (Use Adapter classes).

Source Code:

```
import java.awt.*; import java.applet.*; import java.awt.event.*;
/*<applet code="MouseDemo" width=300 height=300>
</applet>*/
public class MouseDemo extends Applet implements MouseListener, MouseMotionListener
{
int mx = 0;
int my = 0; String msg = "";

public void init() {
addMouseListener(this); addMouseMotionListener(this);
}
public void mouseClicked(MouseEvent me) {mx = 20;
my = 40;
msg = "Mouse Clicked";repaint();
}
public void mousePressed(MouseEvent me) {mx = 30;
my = 60;
msg = "Mouse Pressed";repaint();
}
public void mouseReleased(MouseEvent me) {mx = 30;
my = 60;
msg = "Mouse Released";repaint();
}
public void mouseEntered(MouseEvent me) {mx = 40;
my = 80;
msg = "Mouse Entered";repaint();
}
public void mouseExited(MouseEvent me) {mx = 40;
my = 80;
msg = "Mouse Exited";repaint();
}
public void mouseDragged(MouseEvent me) {mx = me.getX();
my = me.getY();
showStatus("Currently mouse dragged" + mx + " " + my);
repaint();
}
public void mouseMoved(MouseEvent me) {mx = me.getX();
my = me.getY();
showStatus("Currently mouse is at" + mx + " " + my);repaint();
}
public void paint(Graphics g) { g.drawString("Handling Mouse Events", 30, 20);
g.drawString(msg, 60, 40);
}
}
```

Output:



```
Handling Mouse Events
      Mouse Entered

Currently mouse is at388 202
```

Week 11:

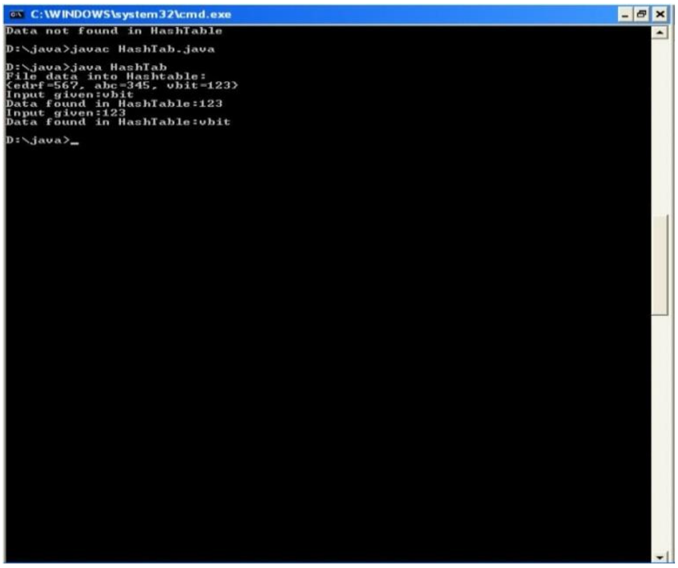
Write a java program that loads names and phone numbers from a text file where the data is organized as one line per record and each field in a record are separated by a tab (\t).it takes a name or phone number as input and prints the corresponding other value from the hash table(hint: use hash tables)

Source Code:

```
import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import
java.io.FileReader;
import
java.io.IOException;
import
java.util.Hashtable;
import
java.util.Iterator;
import java.util.Set;
public class HashTab {
public static void main(String[] args) {
HashTab prog11 = new HashTab();
Hashtable<String, String> hashData = prog11.readFromFile("HashTab.txt");
System.out.println("File data into Hashtable:\n" + hashData);
prog11.printTheData(hashData, "raja");
prog11.printTheData(hashData, "123");
prog11.printTheData(hashData, "---------------------------------------- ");
}
private void printTheData(Hashtable<String, String> hashData, String input)
{
String output = null;
if (hashData != null) {
Set<String> keys = hashData.keySet();
if (keys.contains(input)) {
output = hashData.get(input);
} else {
Iterator<String> iterator = keys.iterator();
while (iterator.hasNext()) {
String key = iterator.next(); String value = hashData.get(key);if
(value.equals(input)) {
output = key;
break;
                                                  }
                                         }
```

```
                                        }
                              }
System.out.println("Input given:" + input);
if (output != null) {
System.out.println("Data found in HashTable:" + output);
} else {
System.out.println("Data not found in HashTable");
}
}
private Hashtable<String, String> readFromFile(String fileName) { Hashtable<String,
String> hashData = new Hashtable<String, String>();try {
File f = new File("D:\\java\\" + fileName); BufferedReader br = new
BufferedReader(new FileReader(f));String line = null;
while ((line = br.readLine()) != null) {
String[] details = line.split("\t"); hashData.put(details[0],
details[1]);
}
} catch (FileNotFoundException e) {
e.printStackTrace();
} catch (IOException e) {
e.printStackTrace();
}
return hashData;
}
}
```

Output:



```
Data not found in HashTable

D:\java>javac HashTab.java

D:\java>java HashTab
File data into Hashtable:
{edrf=567, abc=345, vbit=123}
Input given:vbit
Data found in HashTable:123
Input given:123
Data found in HashTable:vbit

D:\java>_
```

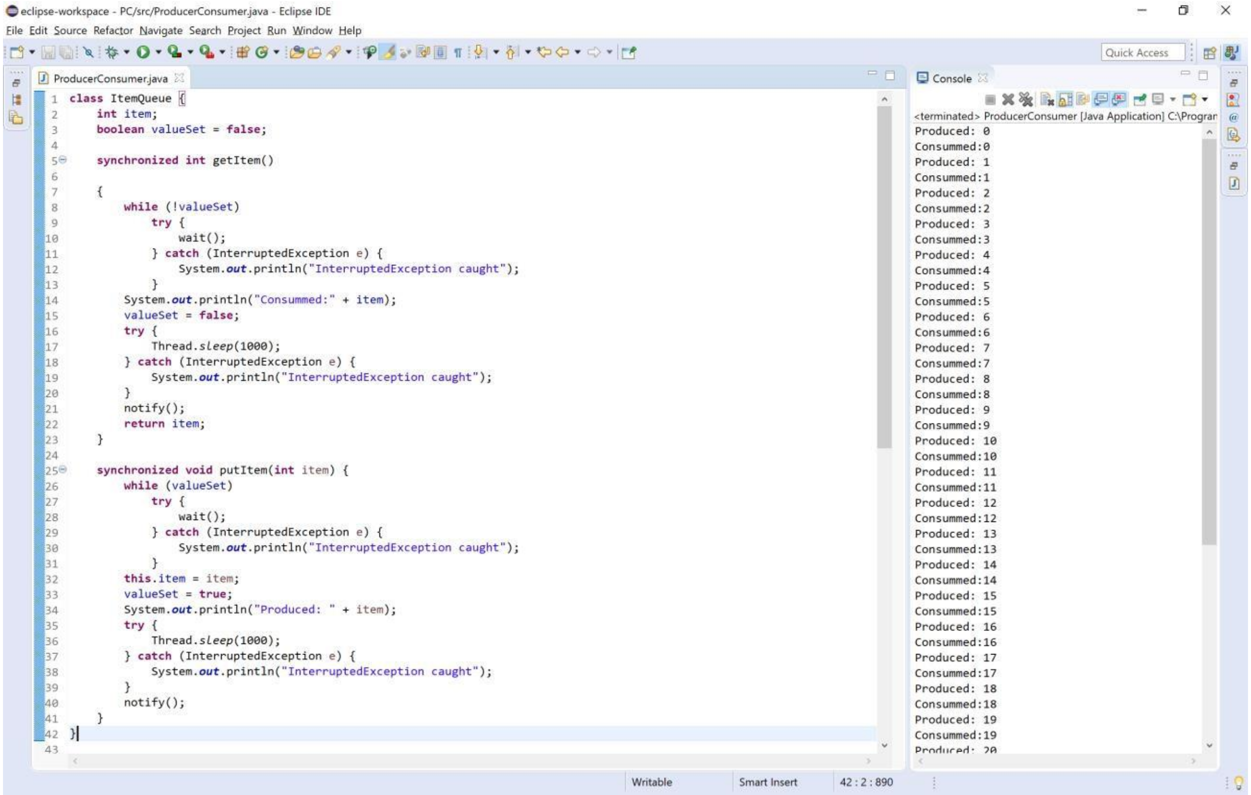
Week – 12

Write A Java Program That Correctly Implements The Producer–Consumer Problem Using The Concept OfInterthread Communication.

Source Code:

```
class ItemQueue {
int item;
boolean valueSet = false;
synchronized int getItem()
{
while (!valueSet)
try {
wait();
} catch (InterruptedException e)
{
System.out.println("InterruptedException caught");
}
System.out.println("Consummed:" + item);
valueSet = false;
try
{
Thread.sleep(1000);
}
catch (InterruptedException e)
{
System.out.println("InterruptedException caught");
}
notify();
return item;
}
synchronized void putItem(int item)
{
while (valueSet)
try {
wait();
} catch (InterruptedException e)
{
System.out.println("InterruptedException caught");
}
this.item = item;
valueSet = true;
System.out.println("Produced: " + item);
try {
Thread.sleep(1000);
} catch (InterruptedException e)
{
System.out.println("InterruptedException caught");
}
notify();
}
}
class Producer implements Runnable
{
 ItemQueue itemQueue;
Producer(ItemQueue itemQueue){
this.itemQueue = itemQueue;
new Thread(this, "Producer").start();
}
public void run() {
int i = 0;
while(true) {
itemQueue.putItem(i++);
}
}
}
class Consumer implements Runnable{

ItemQueue itemQueue;
Consumer(ItemQueue itemQueue){
this.itemQueue = itemQueue;
new Thread(this, "Consumer").start();
}
public void run() {
while(true) {
itemQueue.getItem();
}
}
}
class ProducerConsumer{
public static void main(String args[])
{ ItemQueue itemQueue = new ItemQueue();
new Producer(itemQueue);
new Consumer(itemQueue);
}
}
```

Output:



eclipse-workspace - PC/src/ProducerConsumer.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
ProducerConsumer.java
Console
Writable
Smart Insert
42 : 2 : 890

```java
class ItemQueue {
    int item;
    boolean valueSet = false;

    synchronized int getItem()

    {
        while (!valueSet)
            try {
                wait();
            } catch (InterruptedException e) {
                System.out.println("InterruptedException caught");
            }
        System.out.println("Consummed:" + item);
        valueSet = false;
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            System.out.println("InterruptedException caught");
        }
        notify();
        return item;
    }

    synchronized void putItem(int item) {
        while (valueSet)
            try {
                wait();
            } catch (InterruptedException e) {
                System.out.println("InterruptedException caught");
            }
        this.item = item;
        valueSet = true;
        System.out.println("Produced: " + item);
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            System.out.println("InterruptedException caught");
        }
        notify();
    }
}
```

```
<terminated> ProducerConsumer [Java Application] C:\Progran
Produced: 0
Consummed:0
Produced: 1
Consummed:1
Produced: 2
Consummed:2
Produced: 3
Consummed:3
Produced: 4
Consummed:4
Produced: 5
Consummed:5
Produced: 6
Consummed:6
Produced: 7
Consummed:7
Produced: 8
Consummed:8
Produced: 9
Consummed:9
Produced: 10
Consummed:10
Produced: 11
Consummed:11
Produced: 12
Consummed:12
Produced: 13
Consummed:13
Produced: 14
Consummed:14
Produced: 15
Consummed:15
Produced: 16
Consummed:16
Produced: 17
Consummed:17
Produced: 18
Consummed:18
Produced: 19
Consummed:19
Produced: 20
```

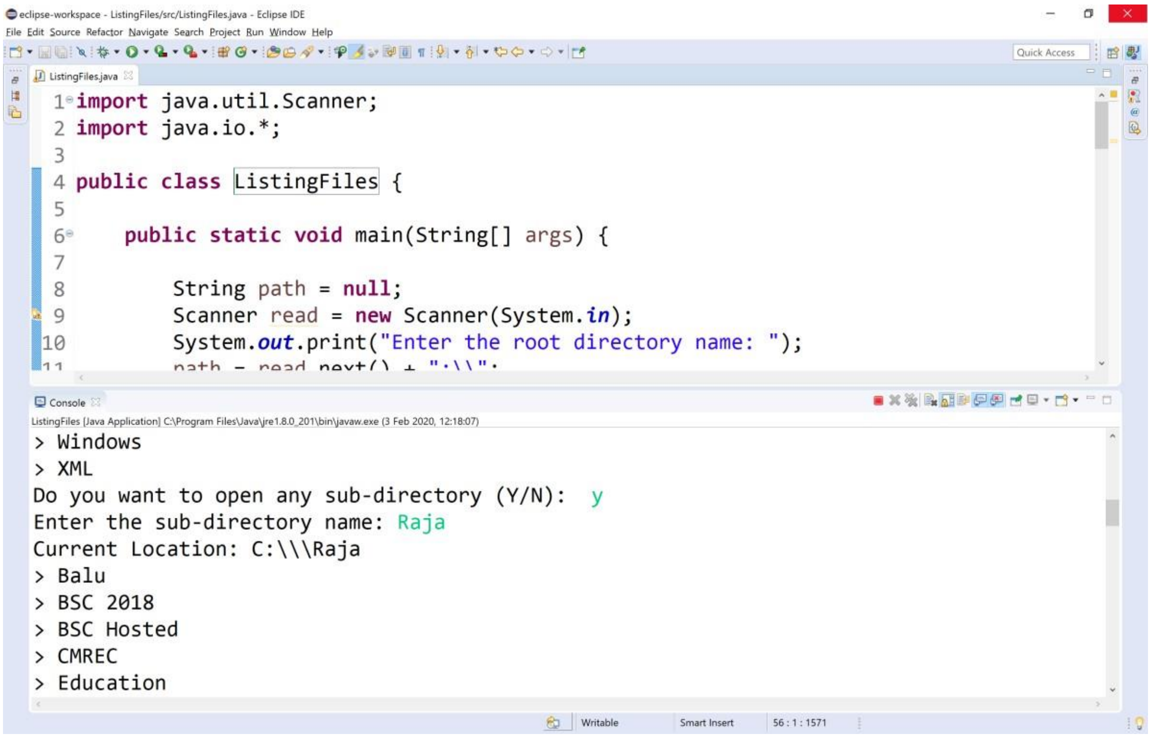
Week – 13

Write a Java program to list all the files in a directory including the files present in all its subdirectories.

Source Code:

```
import java.util.Scanner;
import java.io.*;
public class ListingFiles {
public static void main(String[] args)
{
(Y/N):   ");
exists!");
String path = null;
Scanner read = new Scanner(System.in);
System.out.print("Enter the root directory name: ");
path = read.next() + ":\\";
File f_ref = new File(path);
if (!f_ref.exists()) {
printLine();
System.out.println("Root directory does not exists!");
printLine();
}
 else
{
String ch = "y";
while (ch.equalsIgnoreCase("y"))
 {
printFiles(path);
System.out.print("Do you want to open any sub-directory
ch = read.next().toLowerCase();
if (ch.equalsIgnoreCase("y"))
{
System.out.print("Enter the sub-directory name: ");
path = path + "\\\\" + read.next();
File f_ref_2 = new File(path);
if (!f_ref_2.exists())
{
printLine();
System.out.println("The sub-directory does not
printLine();
int lastIndex = path.lastIndexOf("\\");
path = path.substring(0, lastIndex);
}
}
}
}
System.out.println("***** Program Closed *****");
}
public static void printFiles(String path)
{
 System.out.println("Current Location: " + path);
File f_ref = new File(path);
File[] filesList = f_ref.listFiles();
for (File file : filesList) {
if (file.isFile())
System.out.println("- " + file.getName());
else
}
}
System.out.println("> " + file.getName());
public static void printLine()
{
System.out.println("…………………………………………….");
}
}
```

Output:


eclipse-workspace - ListingFiles/src/ListingFiles.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
ListingFiles.java
Console
Writable
Smart Insert
56 : 1 : 1571

```
import java.util.Scanner;
import java.io.*;

public class ListingFiles {

    public static void main(String[] args) {

        String path = null;
        Scanner read = new Scanner(System.in);
        System.out.print("Enter the root directory name: ");
        path = read.next() + ":\\";
```

ListingFiles [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (3 Feb 2020, 12:18:07)

```
> Windows
> XML
Do you want to open any sub-directory (Y/N):  y
Enter the sub-directory name: Raja
Current Location: C:\\\Raja
> Balu
> BSC 2018
> BSC Hosted
> CMREC
> Education
```

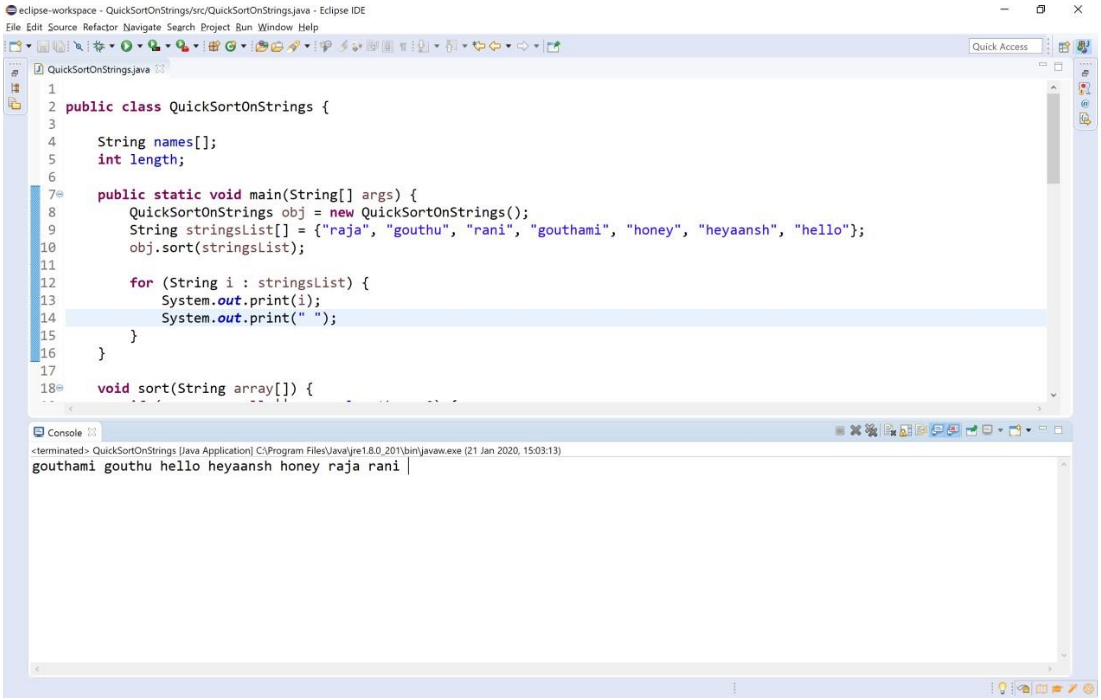
Week 14

Write a Java program that implements Quick sort algorithm for sorting a list of names in ascendingOrder.

Source Code:

```
public class QuickSortOnStrings {

String names[];
int length;

public static void main(String[] args) {
QuickSortOnStrings obj = new QuickSortOnStrings();
String stringsList[] = {"raja", "gouthu", "rani", "gouthami", "honey","heyaansh",
"hello"};
obj.sort(stringsList);

for (String i : stringsList) {
System.out.print(i);
System.out.print(" ");
}
}
void sort(String array[]) {
if (array == null || array.length == 0) {
return;
}
this.names = array; this.length =
array.length;quickSort(0, length -
1);
}
void quickSort(int lowerIndex, int higherIndex) {
int i = lowerIndex;
int j = higherIndex;
String pivot = this.names[lowerIndex + (higherIndex - lowerIndex) / 2];

while (i <= j) {
while (this.names[i].compareToIgnoreCase(pivot) < 0) {i++;
}

while (this.names[j].compareToIgnoreCase(pivot) > 0) {j--;
}
if (i <= j) {
exchangeNames(i, j);i++;
j--;
}
}
if (lowerIndex < j) {
quickSort(lowerIndex, j);
}
if (i < higherIndex) { quickSort(i,
higherIndex);
}
}
void exchangeNames(int i, int j) {
String temp = this.names[i];
this.names[i] = this.names[j];
this.names[j] = temp;
                    }
               }
```

Output:


eclipse-workspace - QuickSortOnStrings/src/QuickSortOnStrings.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
QuickSortOnStrings.java
Console

```

public class QuickSortOnStrings {

    String names[];
    int length;

    public static void main(String[] args) {
        QuickSortOnStrings obj = new QuickSortOnStrings();
        String stringsList[] = {"raja", "gouthu", "rani", "gouthami", "honey", "heyaansh", "hello"};
        obj.sort(stringsList);

        for (String i : stringsList) {
            System.out.print(i);
            System.out.print(" ");
        }
    }

    void sort(String array[]) {
```

```
<terminated> QuickSortOnStrings [Java Application] C:\Program Files\Java\jre1.8.0_201\bin\javaw.exe (21 Jan 2020, 15:03:13)
gouthami gouthu hello heyaansh honey raja rani
```

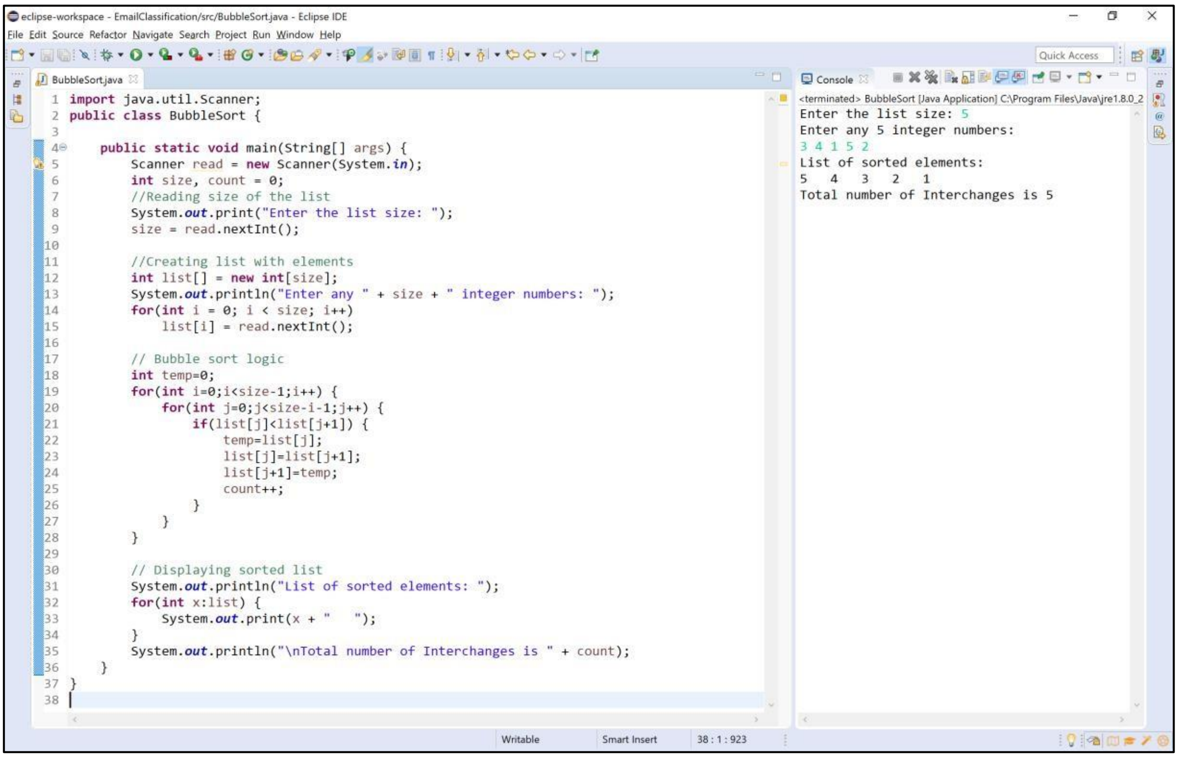
Week 15:

Write a Java program that implements Bubble sort algorithm for sorting in descending order and alsoshows the number of interchanges occurred for the given set of integers.

Source Code:

```
import java.util.Scanner;
public class BubbleSort
{

public static void main(String[] args)
{
Scanner read = new Scanner(System.in);
int size, count = 0;
//Reading size of the list
System.out.print("Enter the list size: ");
size = read.nextInt();

//Creating list with elements
int list[] = new int[size];
System.out.println("Enter any " + size + " integer numbers: ");
for(int i = 0; i < size; i++) list[i] = read.nextInt();

// Bubble sort logic
int temp=0;
for(int i=0;i<size-1;i++) {
for(int j=0;j<size-i-1;j++) {
if(list[j]<list[j+1])
 {
temp=list[j];
 list[j]=list[j+1];
list[j+1]=temp;
count++;
}
}
}
// Displaying sorted list
System.out.println("List of sorted elements: ");
for(int x:list) {
System.out.print(x + "            ");
}
System.out.println("\n Total number of Interchanges is " + count);
}
}
```

Output:


eclipse-workspace - EmailClassification/src/BubbleSort.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
BubbleSort.java
Console
Writable
Smart Insert
38 : 1 : 923

```java
import java.util.Scanner;
public class BubbleSort {

    public static void main(String[] args) {
        Scanner read = new Scanner(System.in);
        int size, count = 0;
        //Reading size of the list
        System.out.print("Enter the list size: ");
        size = read.nextInt();

        //Creating list with elements
        int list[] = new int[size];
        System.out.println("Enter any " + size + " integer numbers: ");
        for(int i = 0; i < size; i++)
            list[i] = read.nextInt();

        // Bubble sort logic
        int temp=0;
        for(int i=0;i<size-1;i++) {
            for(int j=0;j<size-i-1;j++) {
                if(list[j]<list[j+1]) {
                    temp=list[j];
                    list[j]=list[j+1];
                    list[j+1]=temp;
                    count++;
                }
            }
        }

        // Displaying sorted list
        System.out.println("List of sorted elements: ");
        for(int x:list) {
            System.out.print(x + "    ");
        }
        System.out.println("\nTotal number of Interchanges is " + count);
    }
}
```

```
<terminated> BubbleSort [Java Application] C:\Program Files\Java\jre1.8.0_2
Enter the list size: 5
Enter any 5 integer numbers:
3 4 1 5 2
List of sorted elements:
5    4    3    2    1
Total number of Interchanges is 5
```