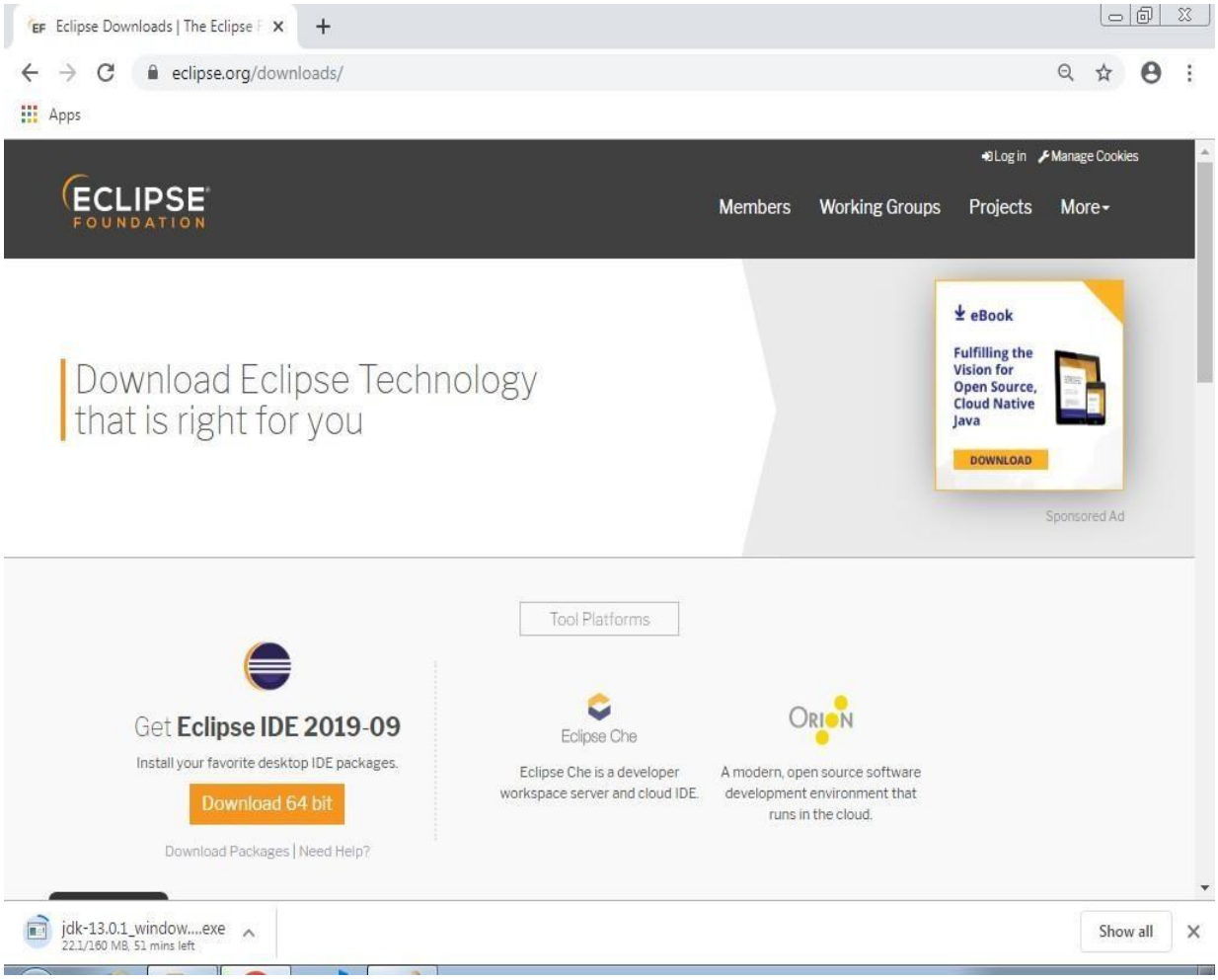
PROGRAMS

Week 1.

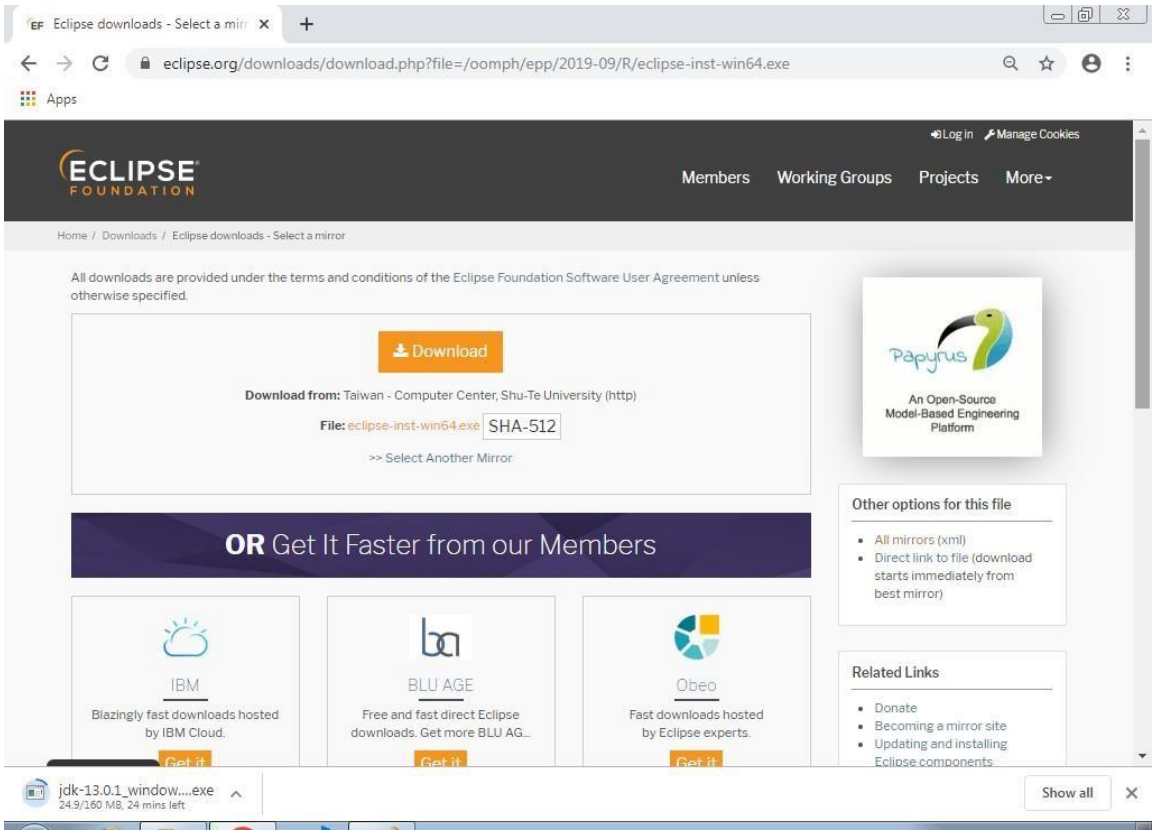
Aim: Use Eclipse or Net bean platform and acquaint with the various menus. Create a test project, add a test class, and run it. See how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods, and classes. Try debug step by step with a small program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

Solution:

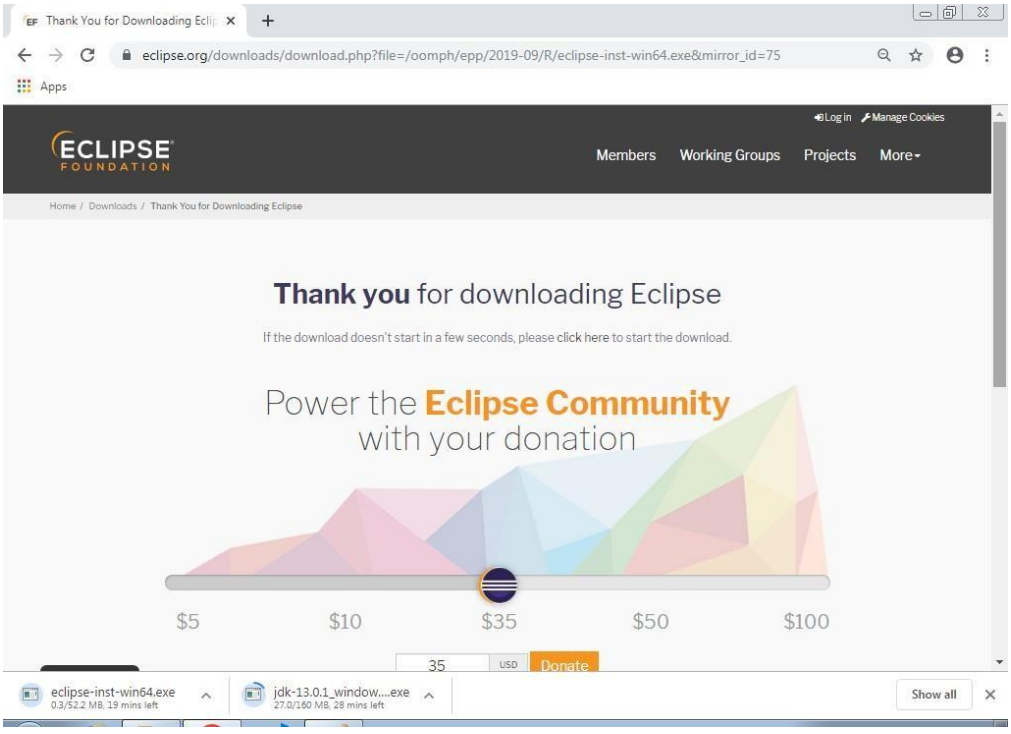
- **Step 1 -** Install JDK in the computer.
- **Step 2 -** Set the path in the Environment Variables from Advanced Setting of computer
- **Step 3 -** Download Eclipse from Eclipse website
- **Step 4 -** Install the Eclipse (follow the screen to install eclipse)



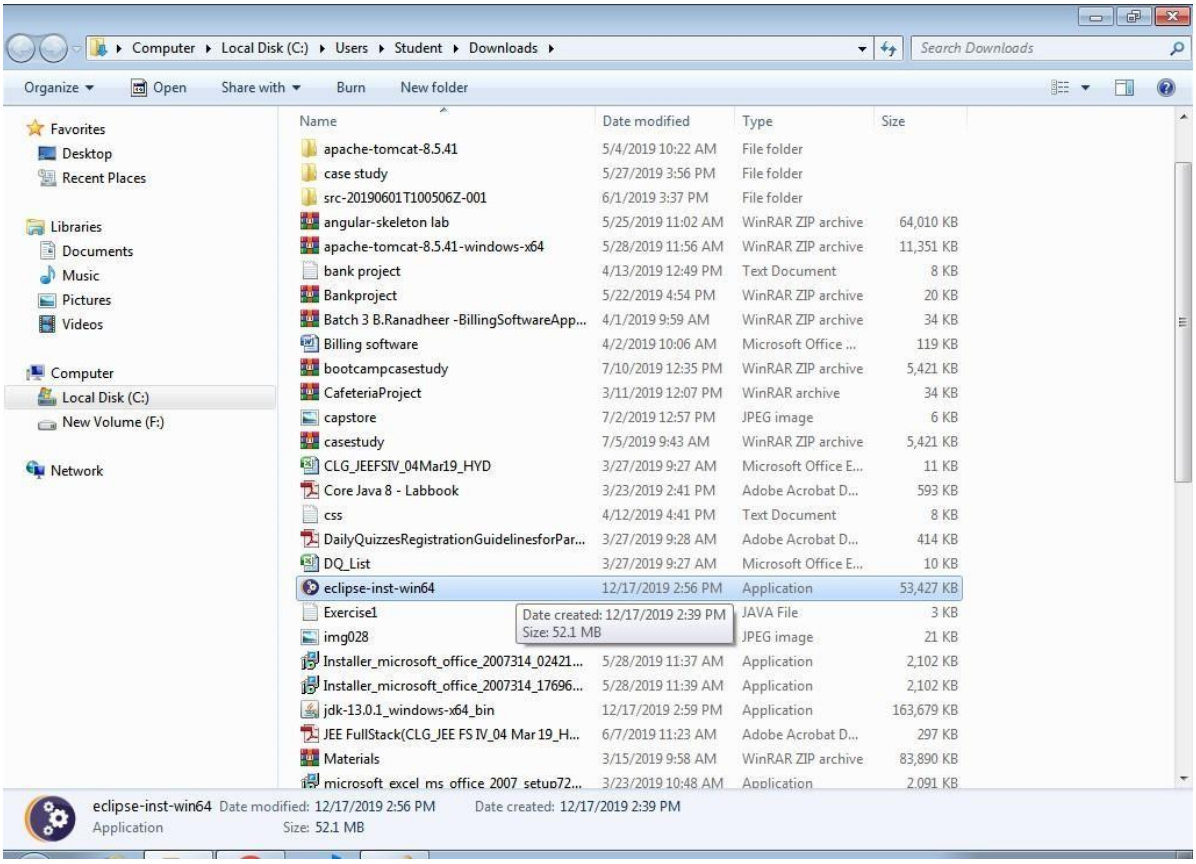
Select the sultable version based on your OS.



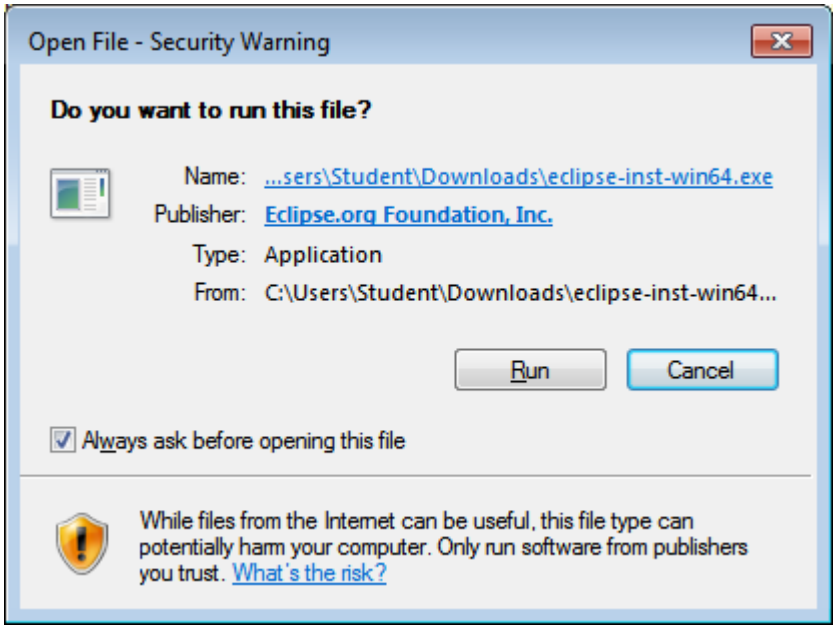
Then download get starts.



Double click on the Eclipse Application.



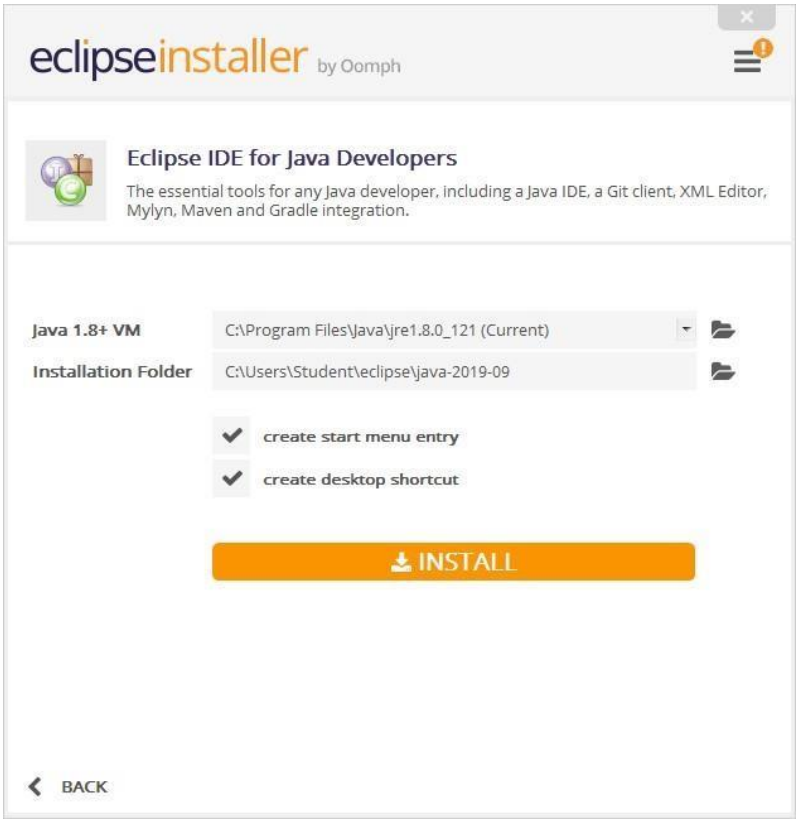
Click on Run in the Security Warning box.



Then, the installation process begins.

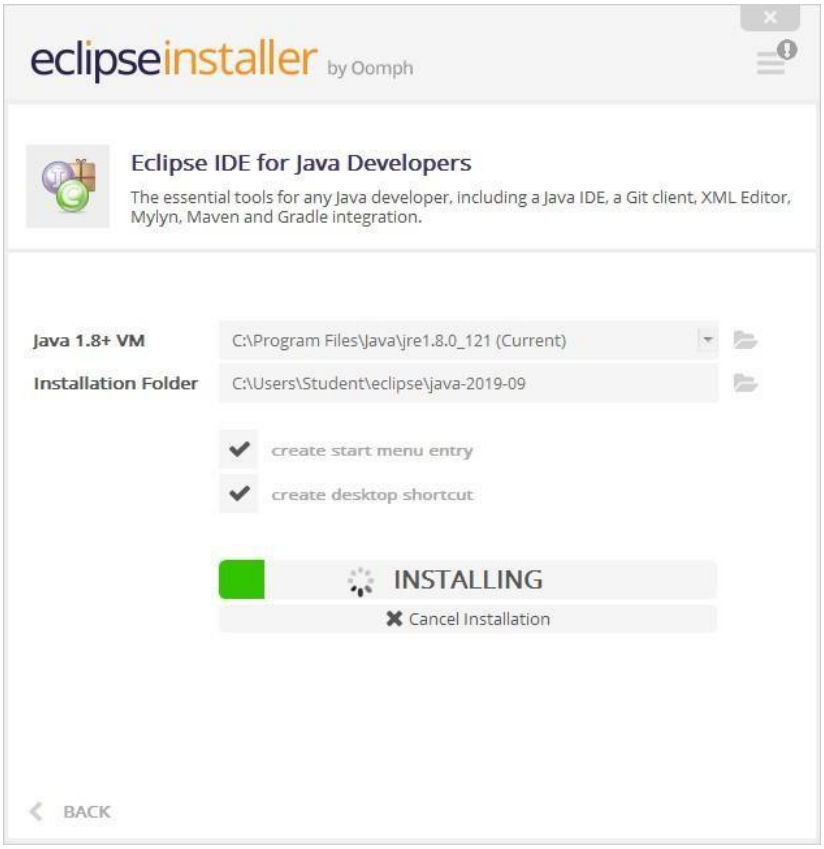
Click on Eclipse IDE for Java Developers.

Click on Install button.

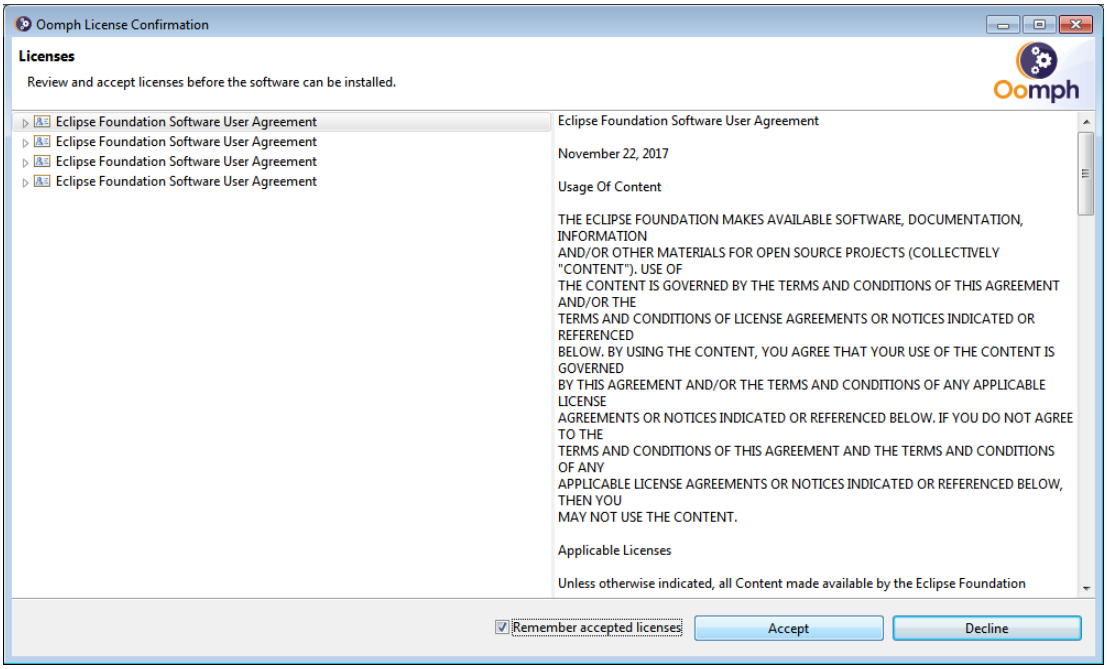


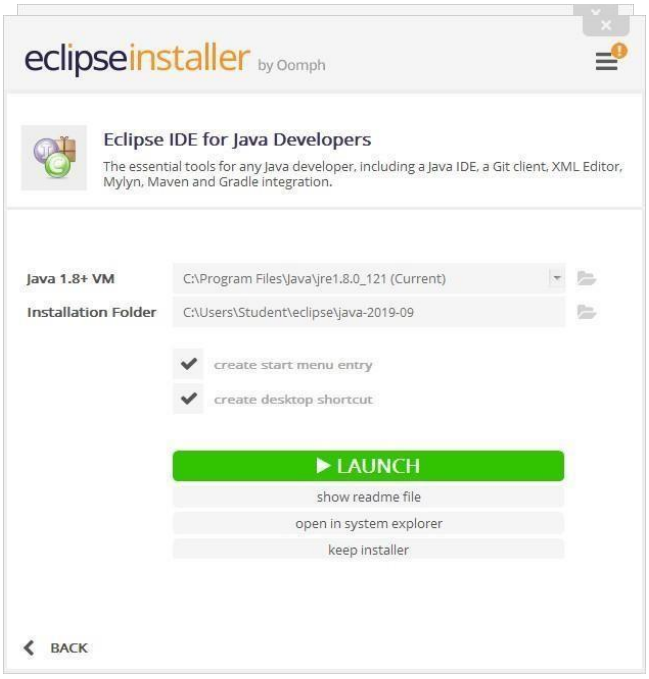
Click on Accept Now.

Then the Eclipse installation begins.

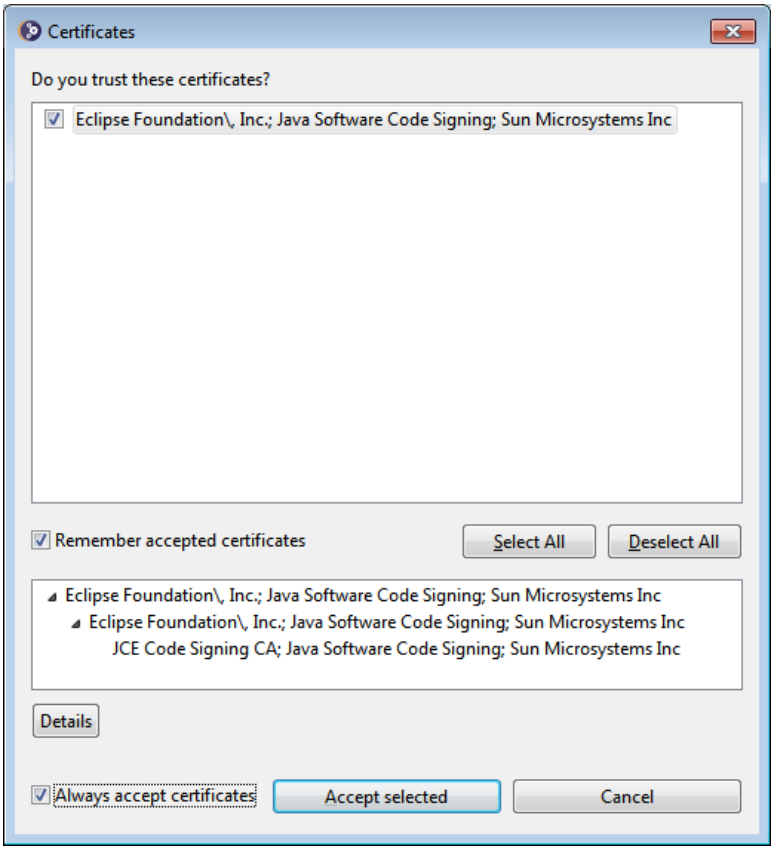


Click on Accept





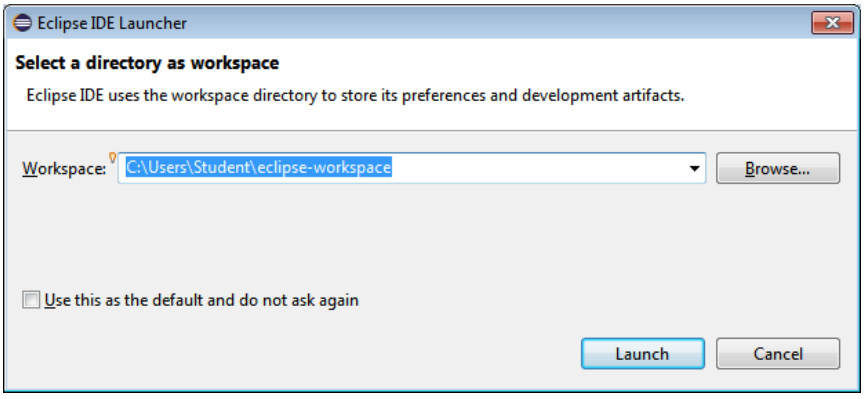
Click on Select All and Accept Selected.



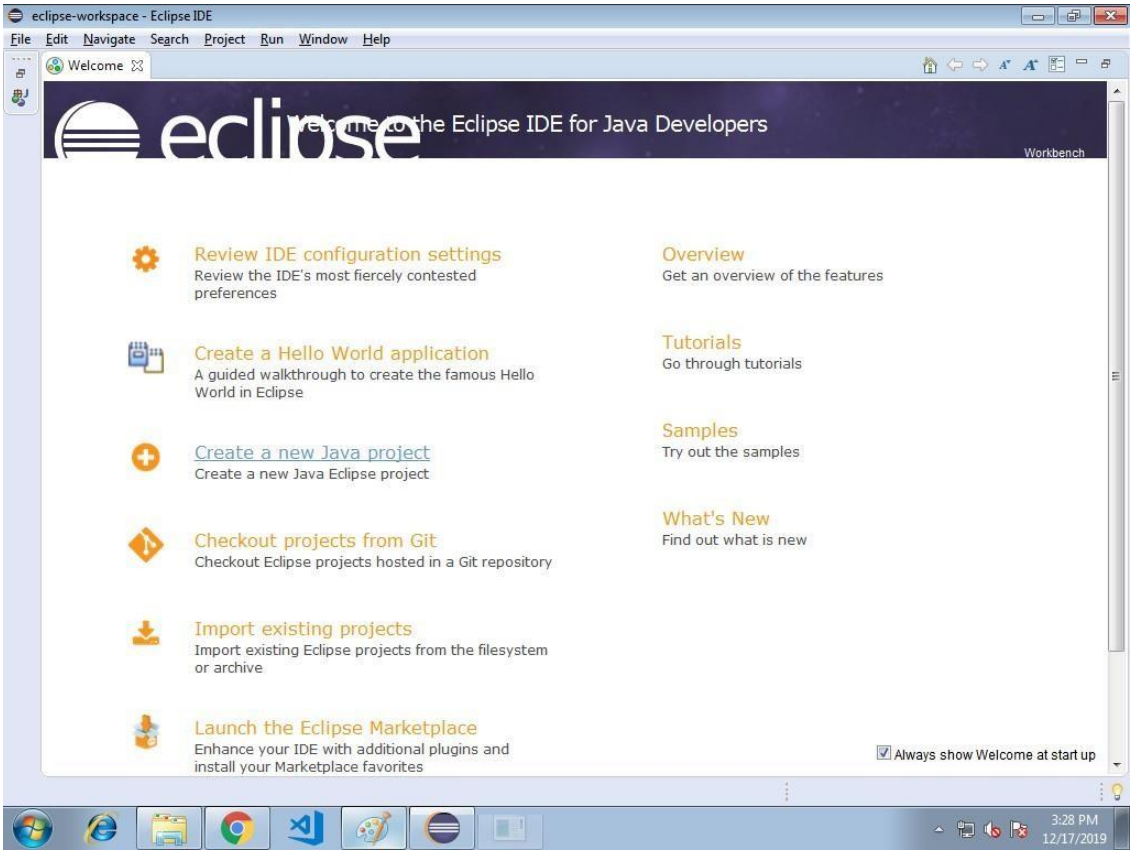
After completing, click on Launch to start the Eclipse IDE.

CREATING PROJECT AND CLASSES IN ECLIPSE IDE

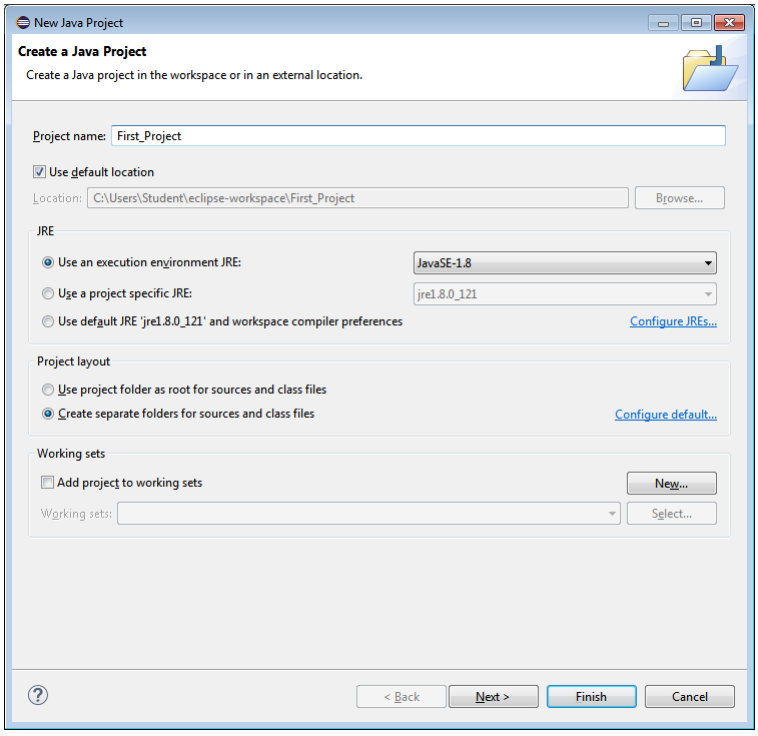
Browse the Workspace for storing the java project and click on Launch.



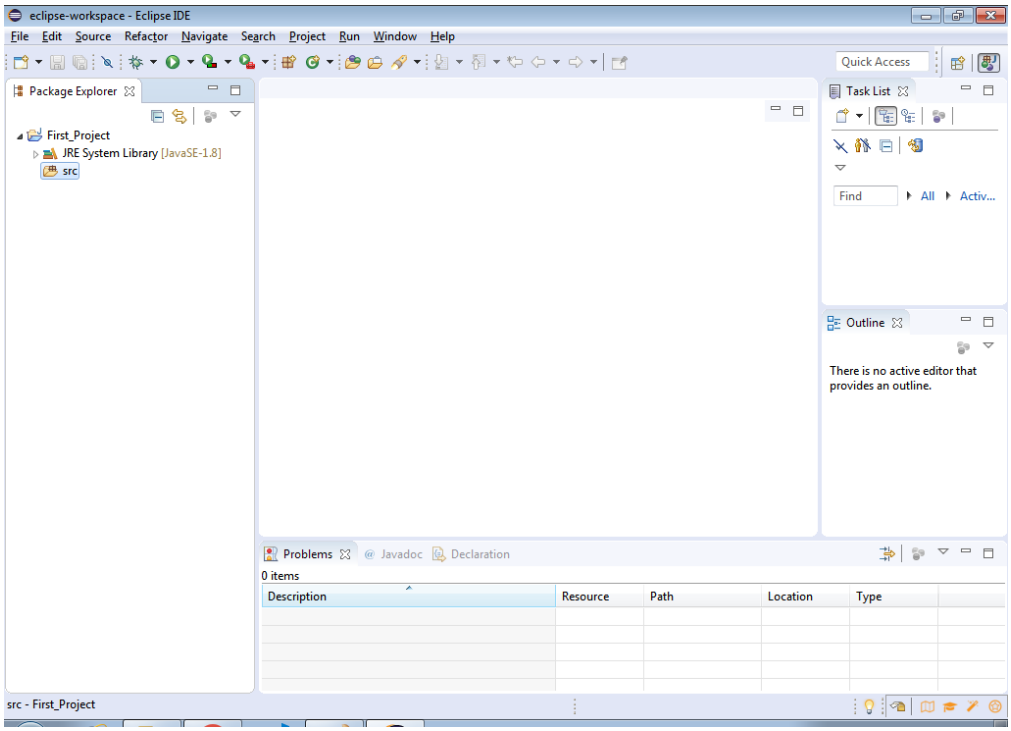
Select "Create a new Java project".

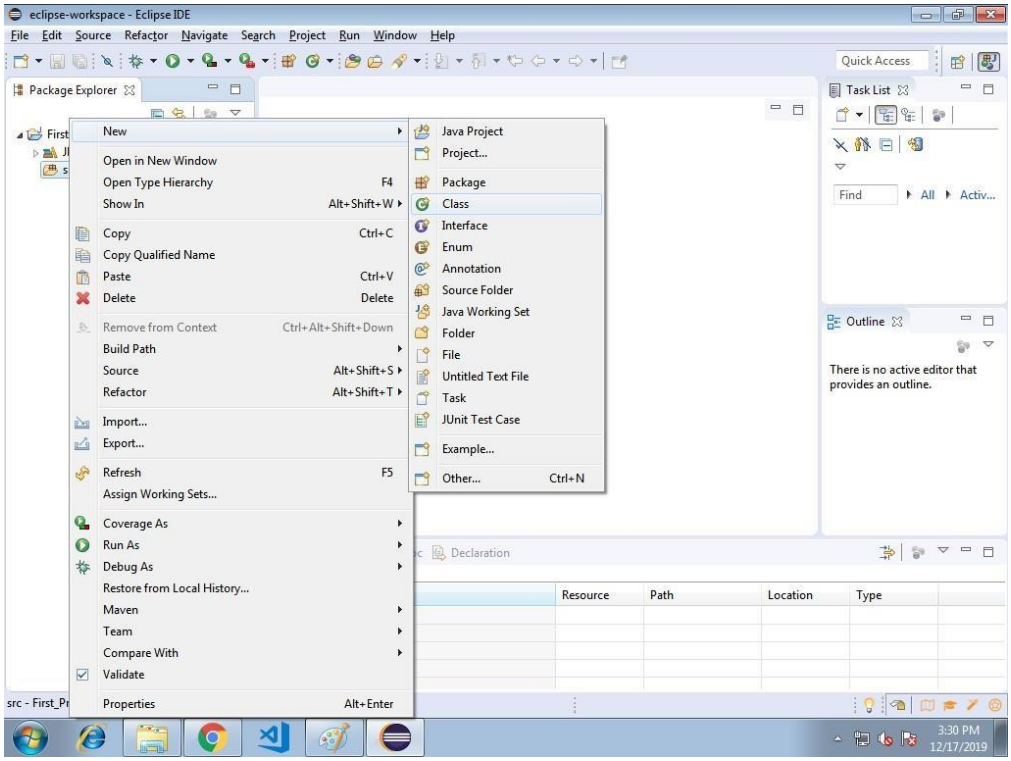


Type the project name and click on Finish.

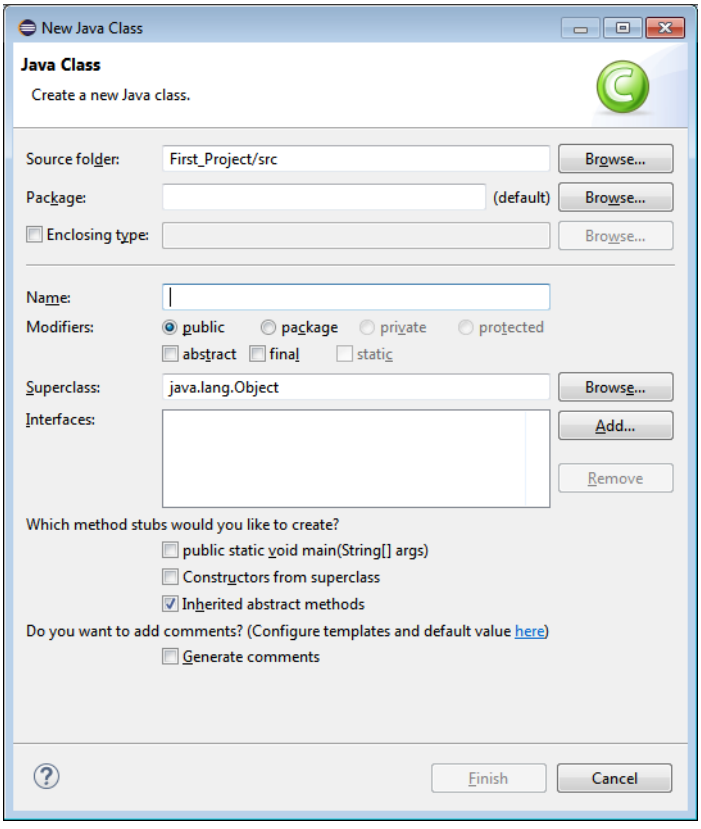


Now, create the class in src directory from Package Explorer window.

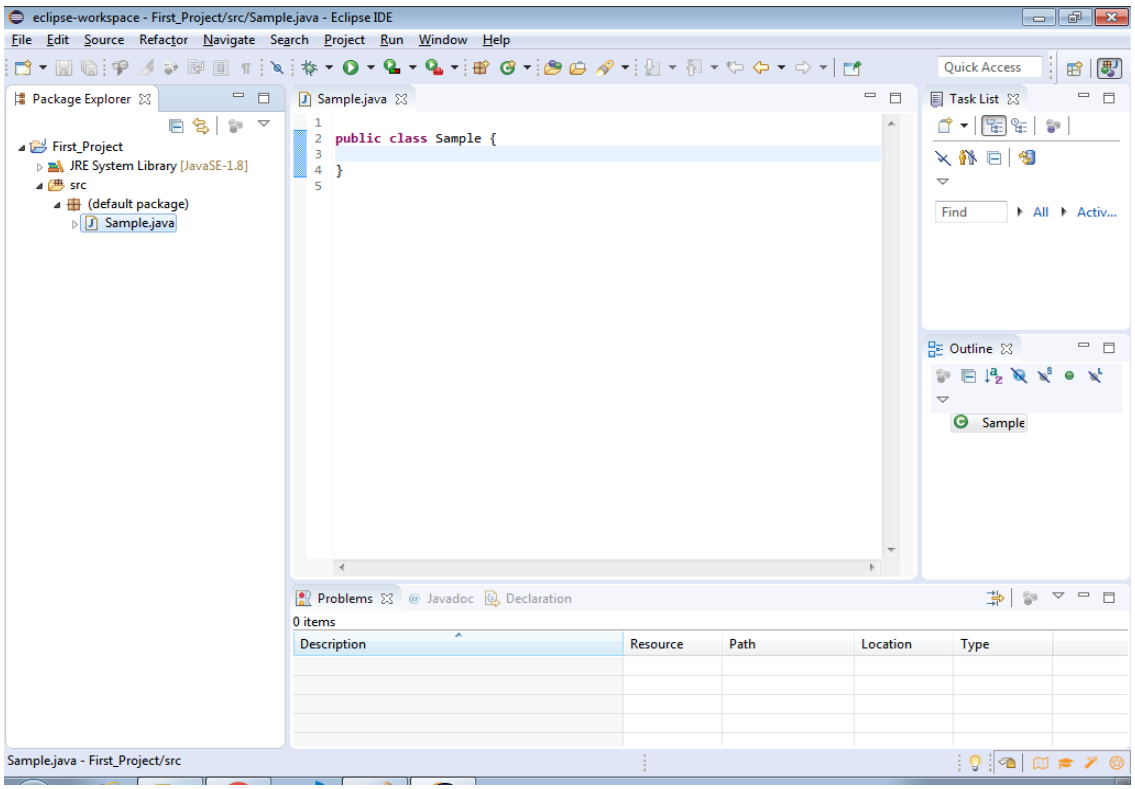




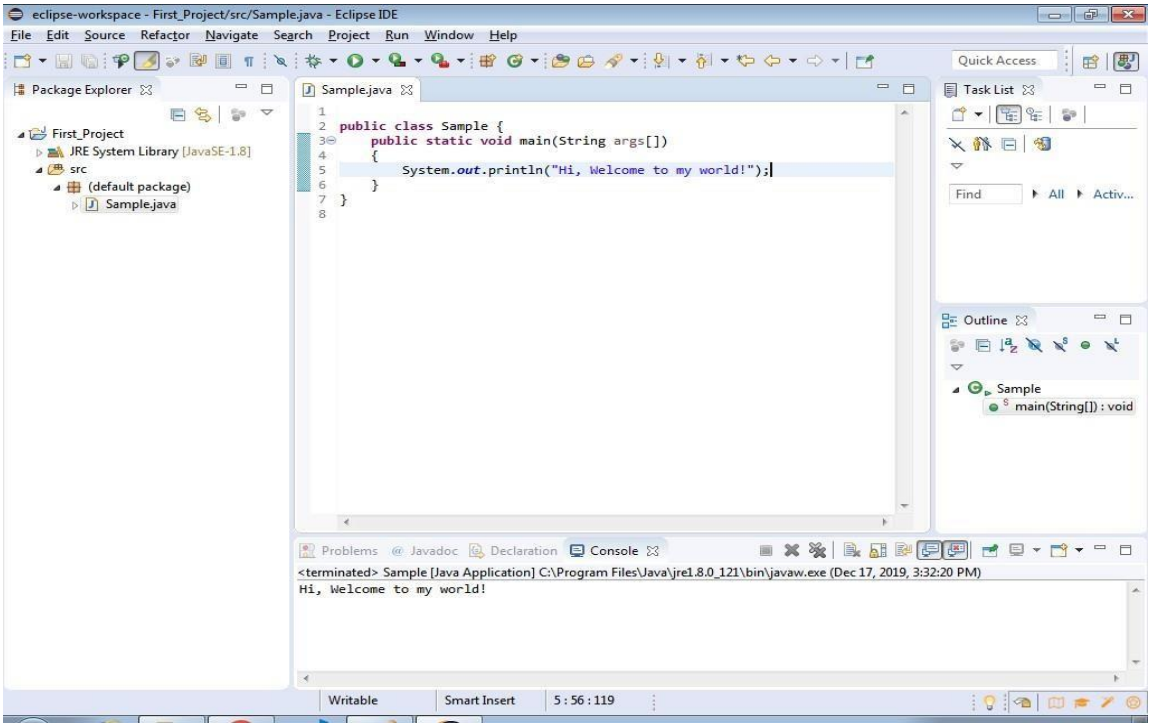
Type the class name and click on Finish.



Type the java code.



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



Program Using If and For Loop:-

```
public class Prog1
{

	public static void main(String[] args)
	{
		System.out.println("\n Prog. is showing
		even no"); for(int i=2;i<=20;i++)
		{
			if(i%2==0)
			{
				System.out.print("\n "+i);
			}

		}
	}
}
```

Compile:-

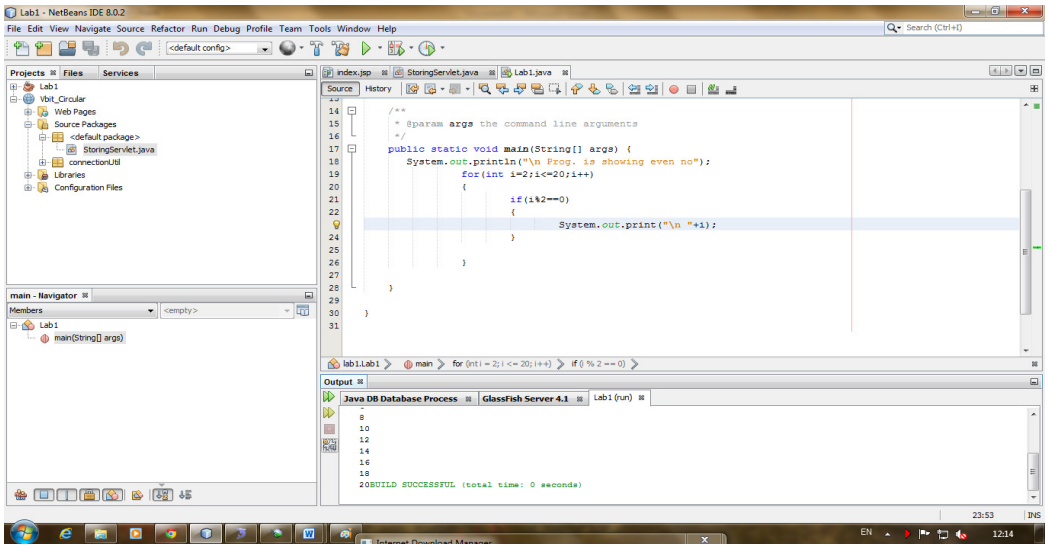
```
D:>javac Prog1.java
```

Run:-

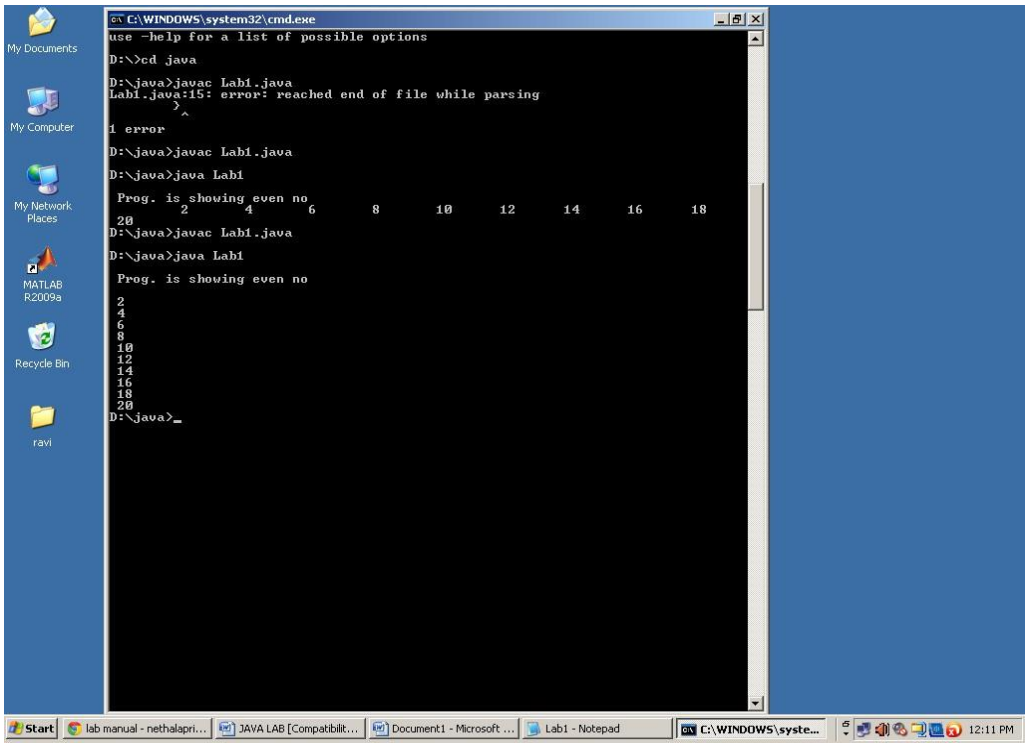
```
D:>java Prog1
```

Output:-

In Netbeans IDE:-



In Command Prompt:-



Viva Questions:

- Explain JDK, JRE and JVM?
- Explain public static void main(String args[]) in Java
- Why Java is platform independent?
- What is an Eclipse IDE?
- What are the system requirements for Eclipse IDE to run successfully on our computersystem?

Week 2:

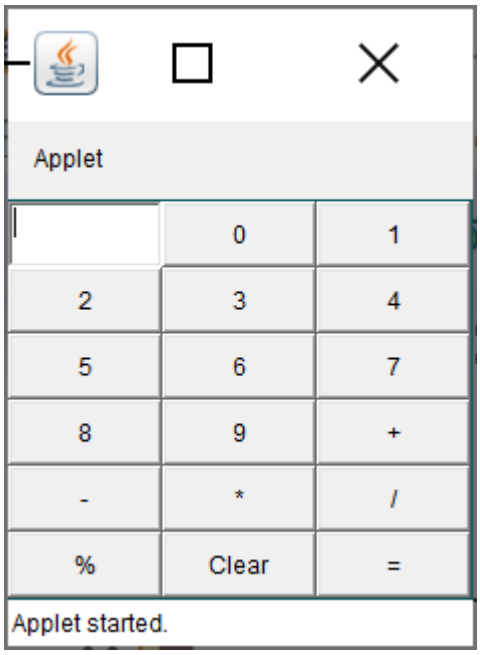
Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -,*, % operations. Add a text field to display the result. Handle any possible exceptions like divided by zero.

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:



Viva Questions:

- What is an Applet ?
- What is the difference between an Applet and a Java Application ?
- Name three Component subclasses that support painting.
- What is the relationship between an event-listener interface and an event-adapter class ?
- How can a GUI component handle its own events ?

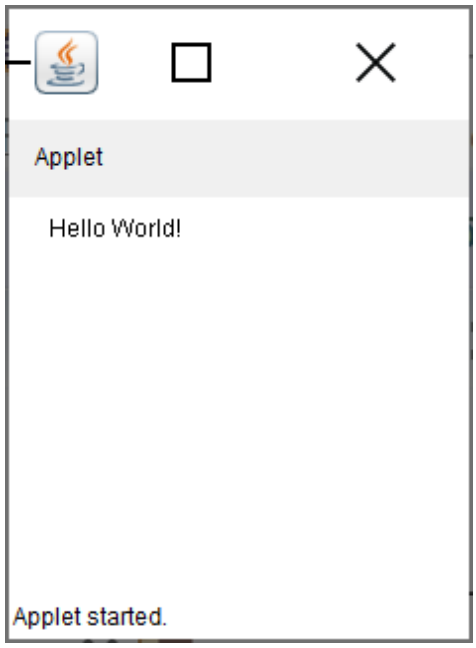
Week 3:

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.

Source code for question a:

```
// 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 Applet1 extends Applet
{
	// Paint method to display the message.
	public void paint(Graphics g)
	{
		g.drawString("Hello World!",20,20);
	}
}
```

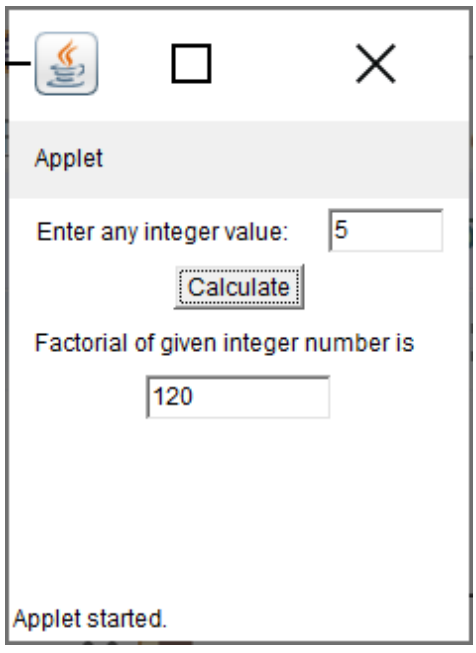
Output:



Source code for question b:

```
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:



Viva Questions:

1. What is Applet?
2. What is the order of method invocation in an Applet
3. How will you initialize an applet?
4. How do I load a serialized applet?
5. Define AWT.

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.

ALGORITHMS:

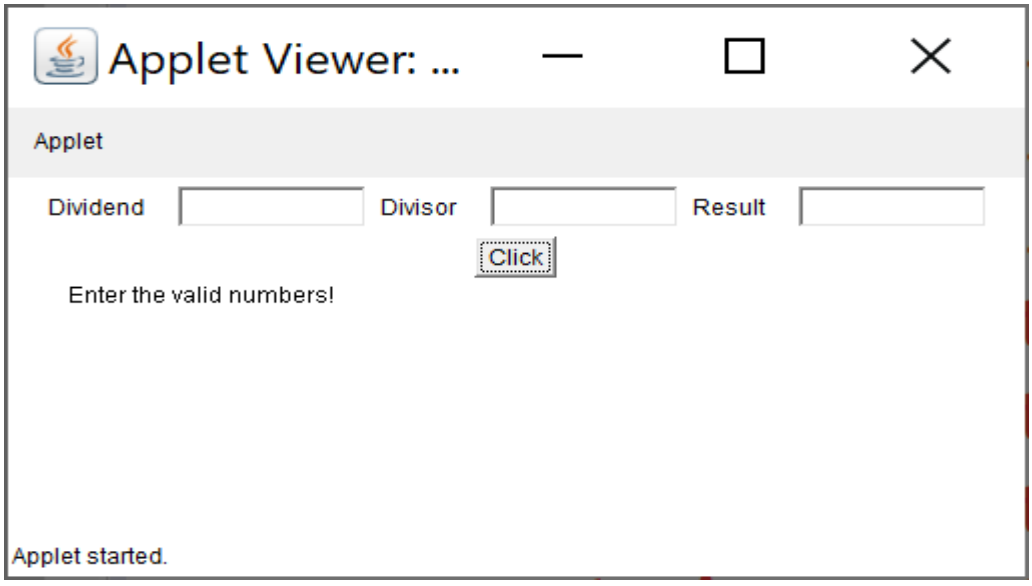
1. create an applet from extending Applet class.
2. Draw a line using drawLine()method.
3. Draw rectangle using drawRec() method.
4. Draw an ovel using drawOvel()method.
5. Include all these methods in the paint()method.

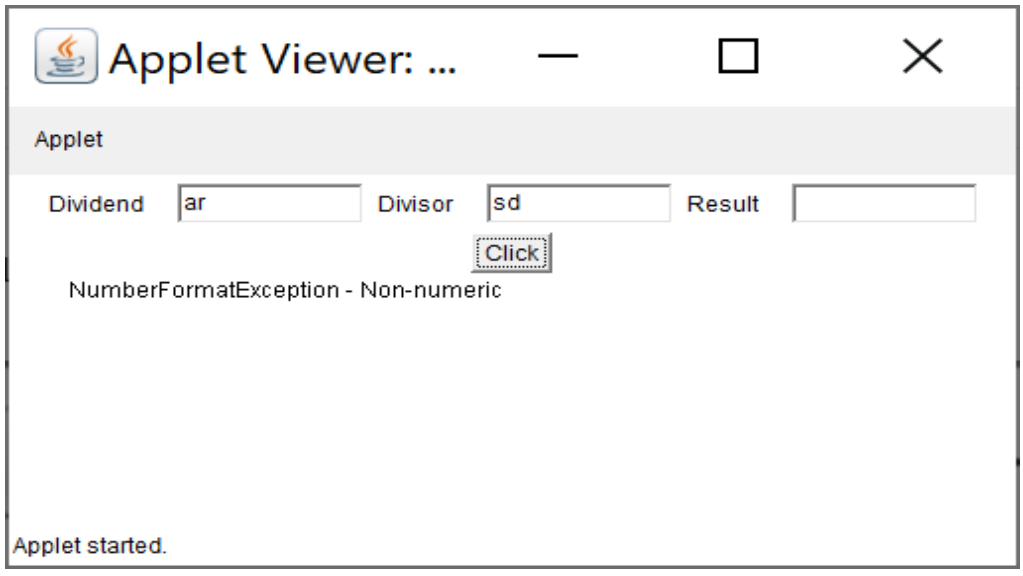
Source code:

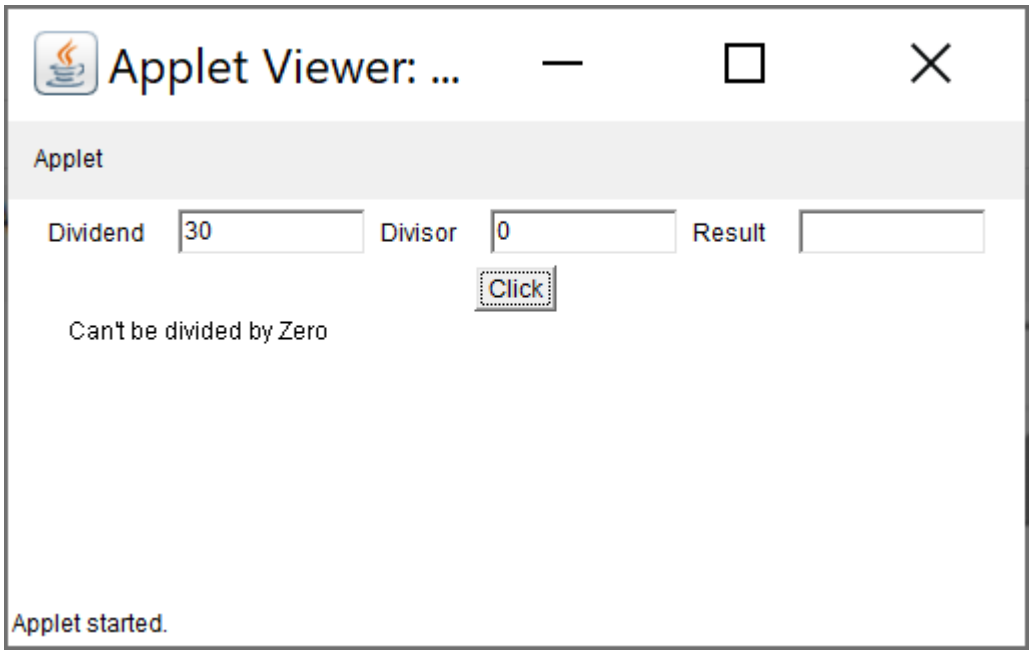
```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;

/*<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;
```

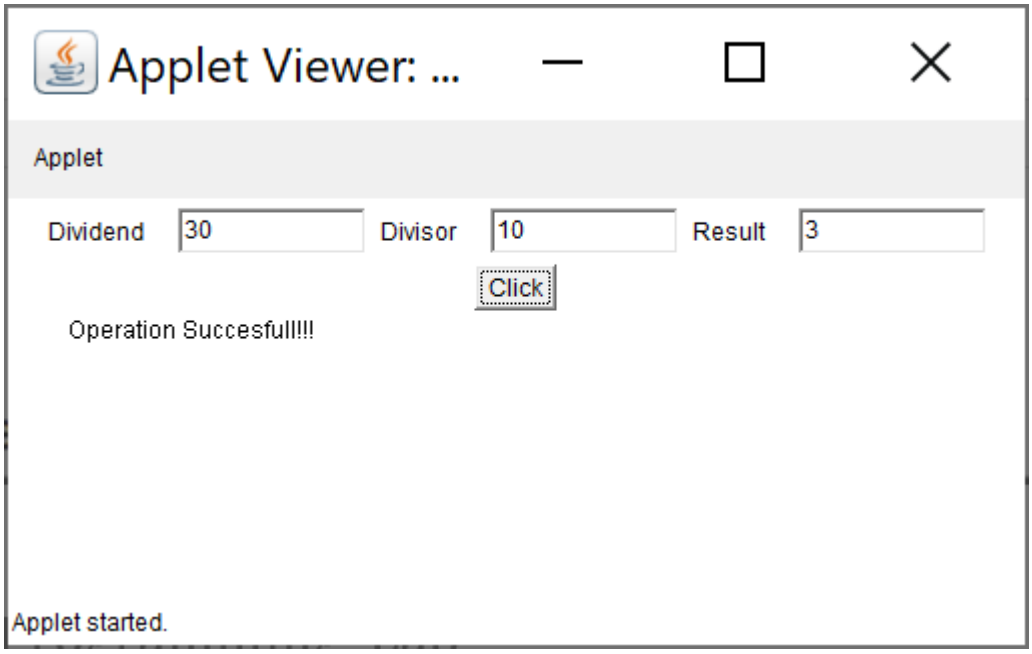
```
                        res.setText(String.valueOf(num3));
                        msg = "Operation Succesfull!!!";
                        repaint();
                    } catch (NumberFormatException ex) {
                        System.out.println(ex);
                        res.setText("");
                        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:









Viva Questions:

1. What is Exceptionhandling?
2. What is the difference between Exception and Error?
3. what is interface?
4. types of exceptions in Java?
5. Write the Syntax for Try … Catch

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.

ALGORITHMS:

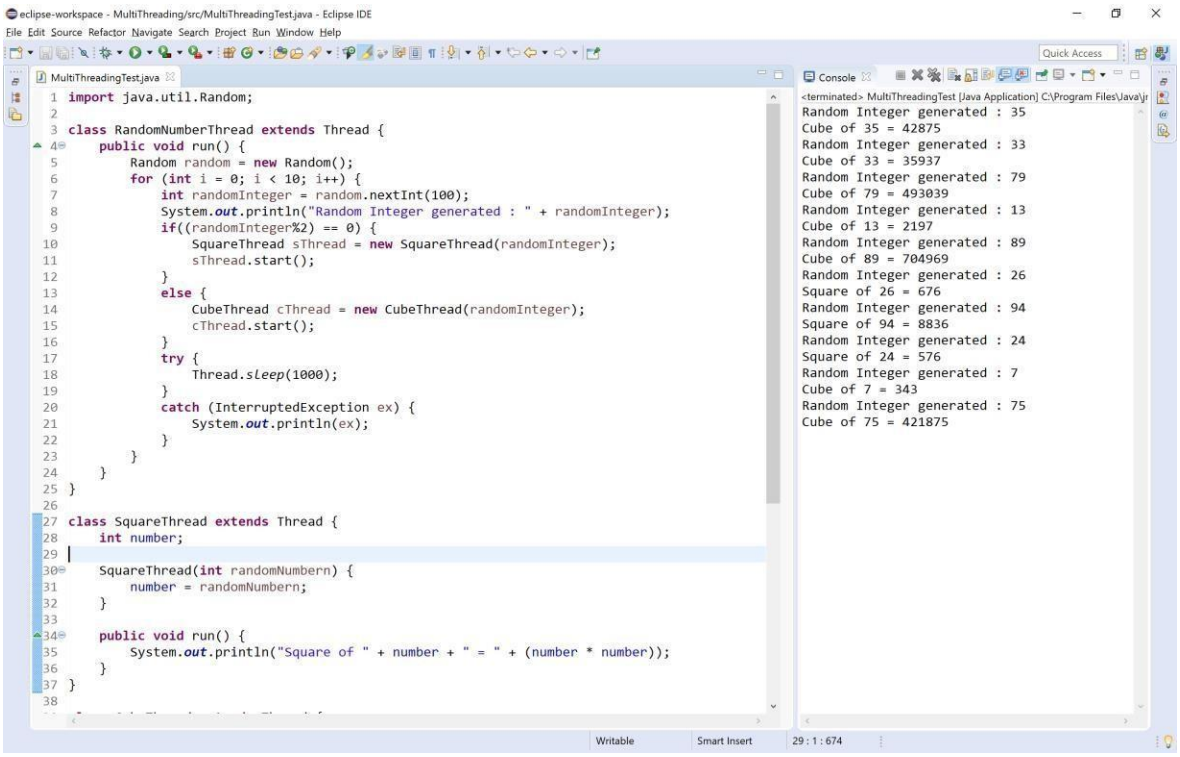
1Create a thread class by implementing the Runnable interface.
2.Start the thread in the constructor.
3.implement the run() method.
4.create multiple threads in the main()
5.Display the threads

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
                  {
                        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));
        }
  }
  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:



Viva Questions:

1. What is thread?
2. Define Random value with ex
3. Is it possible to start a thread twice?
4. When should we interrupt a thread?
5. What is Daemon Thread?

Week 6:

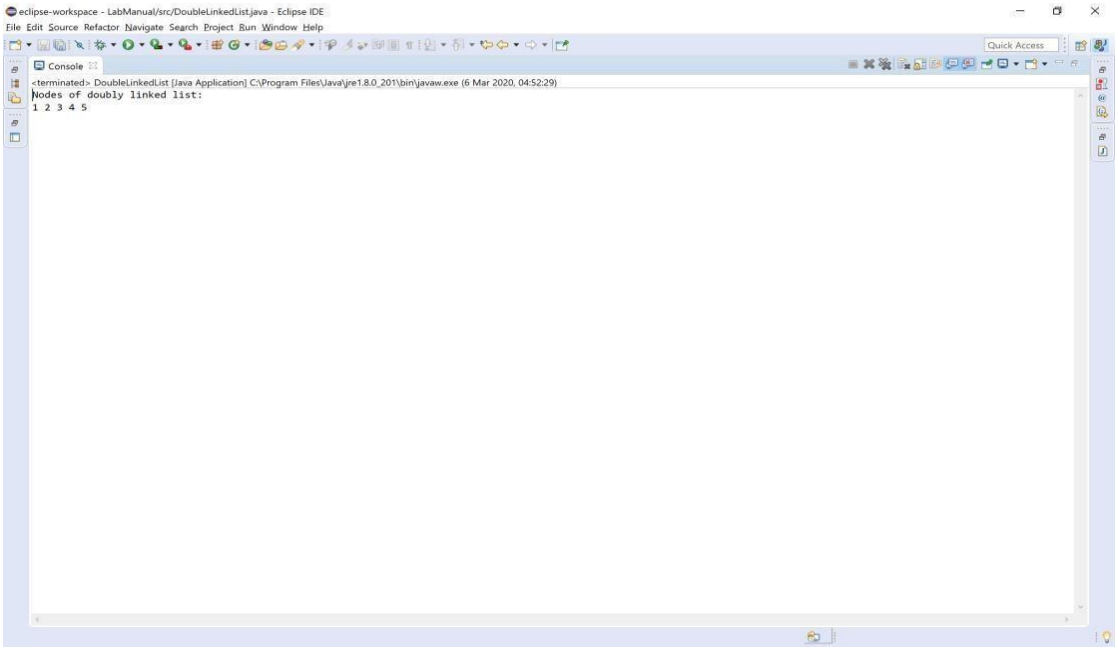
Write a Java program for the following: Create a doubly linked list of elements. Delete a given element from the above list.Display the contents of the list after deletion.

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:



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.

ALGORITHMS:

1. create an applet from extending Applet class.
2. Draw a line using drawLine()method.
3. Draw rectangle using drawRec() method.
4. Draw an ovel using drawOvel()method.
5. Include all these methods in the paint()method.
6. Display the line.rectangle and ovel.

Source code:

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

   //<applet code = "TrafficLights" width = 1000 height = 500></applet>

public class TrafficLights 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:

Viva Questions

1. what is Awt?
2. what is java eventhandling?
3. how to do awt Button Class declaration?
4. what is java Awt Textfield?
5. write the syntax for serFont()

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 AbstractClass {
	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:

```
C:\Users\sriindu\Desktop\sasi>javac AbstractClass.java

C:\Users\sriindu\Desktop\sasi>java AbstractClass

        *** 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: 3 3
        The area of Triangle is: 4

        *** Finding the Area of Cricle ***
        Enter Radius: 3
        The area of Cricle is: 28.26

C:\Users\sriindu\Desktop\sasi>
```

Viva Questions:

1. What is Abstract Class?
2. Write the syntax for Print Area().
3. What is the use of Scanner option
4. Write the syntax for Scanner

WEEK 9:

9). Suppose that a table named Table.txt is stored in a text file. The first line in the file header and the remaining lines correspond to row in the table. The elements are separatedby commas. Write a Java program to display the table using labels in grid layout.

Program:-

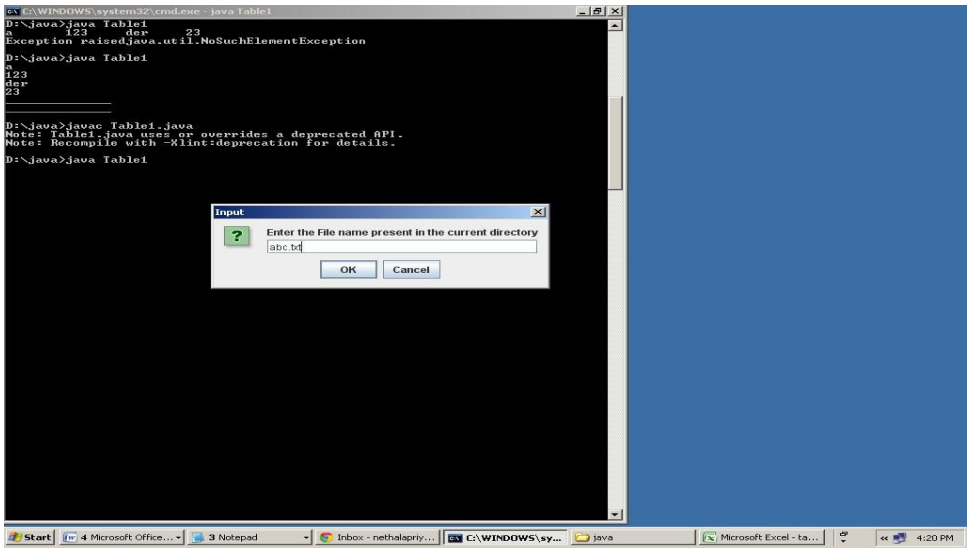
```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import  java.util.*;
import java.io.*;
public class Table1 extends JFrame

{

        int i=0;

        int j=0,k=0;

        Object data[][]=new Object[5][4];

        Object list[][]=new Object[5][4];

        JButton save;

        JTable table1;

        FileInputStream fis;

        DataInputStream dis;

        public Table1()

        {

                String d= " ";

                Container con=getContentPane();

                con.setLayout(new  BorderLayout());

                final String[] colHeads={"Name","Roll Number","Department","Percentage"};

                try

                {

                   String s=JOptionPane.showInputDialog("Enter the File name present in the current
                   directory");FileInputStream fis=new FileInputStream(s);
                   DataInputStream dis = new DataInputStream(fis);

                   while ((d=dis.readLine())!=null)
                   {
                         StringTokenizer st1=new StringTokenizer(d,",");
                         while (st1.hasMoreTokens())
                            {
                            for (j=0;j<4;j++)
                            {
```

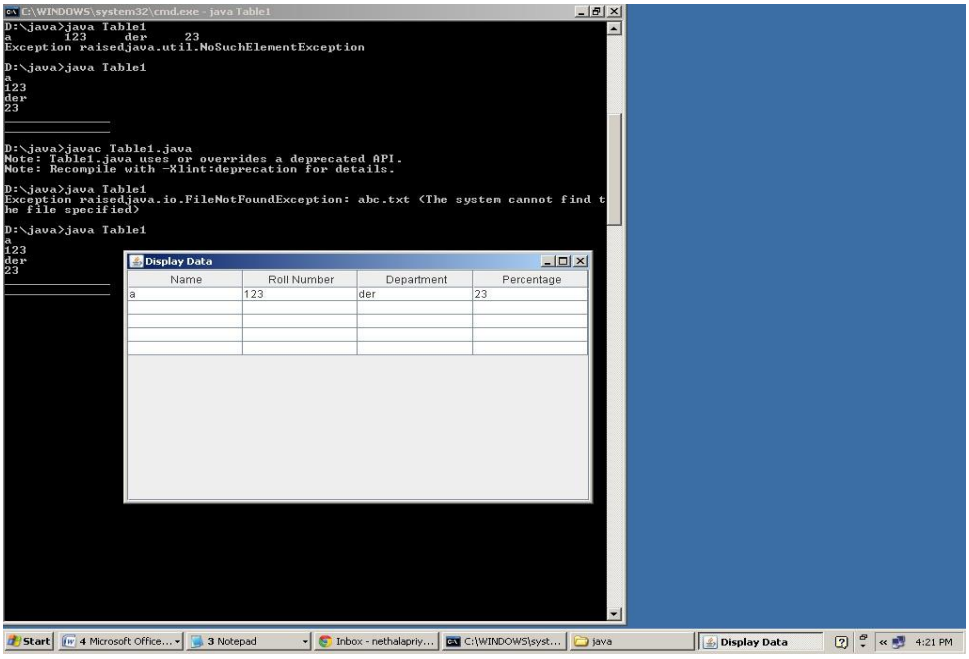
```
                                    data[i][j]=st1.nextToken();
                                    System.out.println(data[i][j]);
                                }
                                i++;
                            }
                             System.out.println ("__");
                        }
                } catch (Exception e)
                  {
                        System.out.println ("Exception raised" +e.toString());
                  }
                  table1=new JTable(data,colHeads);
                  int  v=ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEEDED;
                  int h=ScrollPaneConstants.HORIZONTAL_SCROLLBAR_AS_NEEDED;
                  JScrollPane scroll=new JScrollPane(table1,v,h);
                  con.add(scroll,BorderLayout.CENTER);
            }
      public static void main(String args[])
        {

            Table1 t=new Table1();

            t.setBackground(Color.green);

            t.setTitle("Display Data");

            t.setSize(500,300);

            t.setVisible(true);

            t.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

      }
}
```

Abc.txt:-

a,123,der,23b,456,frg,45

Output:-


C:\WINDOWS\system32\cmd.exe - java Table1
D:\java>java Table1
a 123 der 23
Exception raisedjava.util.MoSuchElementException
D:\java>java Table1
a
123
der
23
D:\java>javac Table1.java
Note: Table1.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
D:\java>java Table1
Input
Enter the File name present in the current directory
abc.txt
OK
Cancel
Start
4 Microsoft Office...
3 Notepad
Inbox - nethalapriy...
C:\WINDOWS\sy...
java
Microsoft Excel - ta...
4:20 PM


C:\WINDOWS\system32\cmd.exe - java Table1
D:\java>java Table1
a 123 der 23
Exception raisedjava.util.MoSuchElementException
D:\java>java Table1
a
123
der
23
D:\java>javac Table1.java
Note: Table1.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
D:\java>java Table1
Exception raisedjava.io.FileNotFoundException: abc.txt (The system cannot find the file specified)
D:\java>java Table1
a
123
der
23
Display Data
Name
Roll Number
Department
Percentage
a
123
der
23
Start
4 Microsoft Office...
3 Notepad
Inbox - nethalapriy...
C:\WINDOWS\syst...
java
Display Data
4:21 PM

Week 10:

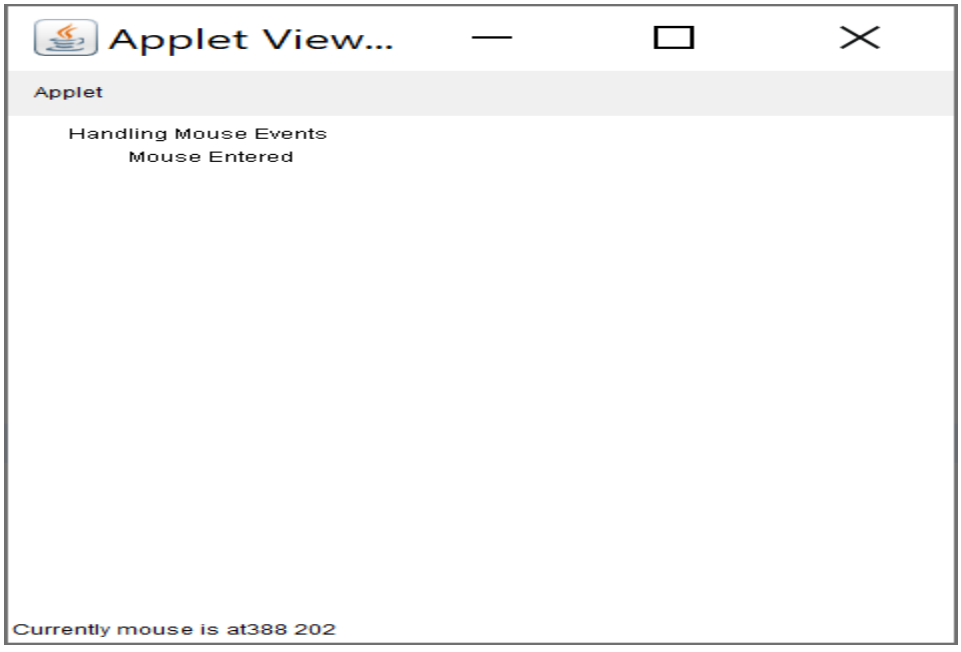
Write a Java program that handles all mouse events and shows the event name at the center of the windowwhen a mouse event is fired (Use Adapter classes).

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:



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("C:\\Users\\sriindu\\Desktop\\sasi\\" + 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;
	}
}
```

HashTab.txt

raja	123
rani	456
anu	789

Output:

C:\Users\sriindu\Desktop\sasi>javac HashTab.java

C:\Users\sriindu\Desktop\sasi>java HashTab

File data into Hashtable:
{anu=789, rani=456, raja=123}
Input given : raja
Data found in HashTable : 123
Input given : 123
Data found in HashTable : raja
Input given:
Data not found in HashTable

Week – 12

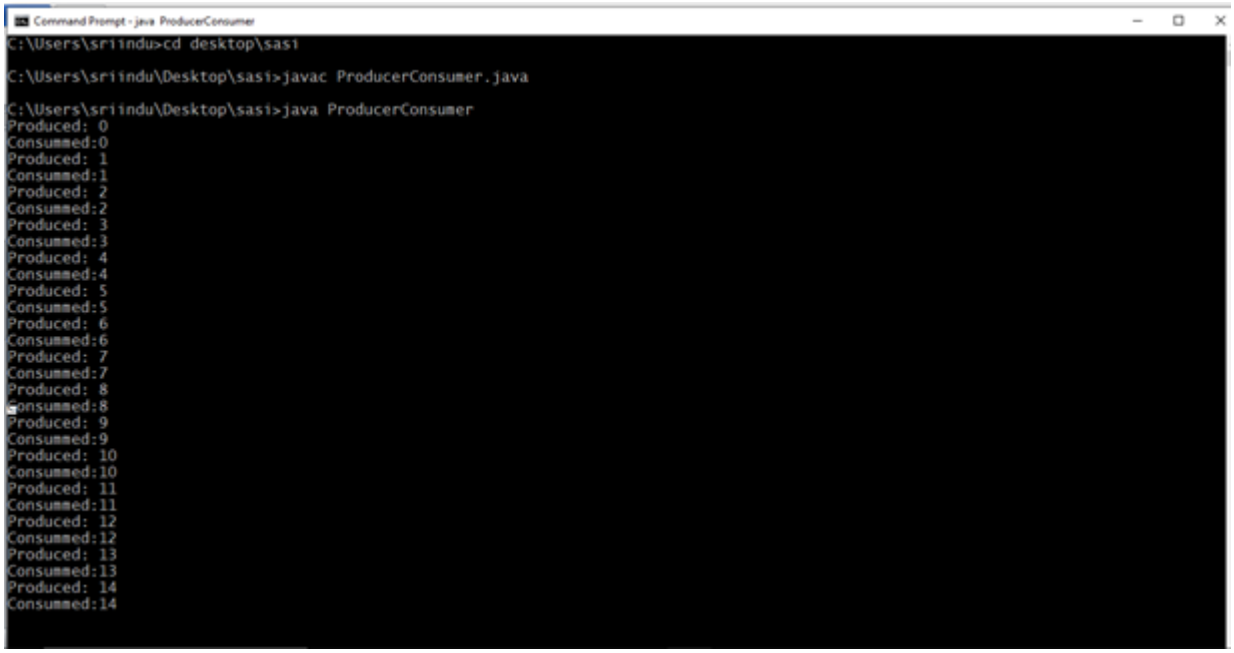
Write a Java program that correctly implements the producer – consumer problem using the concept of interthread communication.

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:


Command Prompt - java ProducerConsumer
C:\Users\sriindu>cd desktop\sasi
C:\Users\sriindu\Desktop\sasi>javac ProducerConsumer.java
C:\Users\sriindu\Desktop\sasi>java ProducerConsumer
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

Week – 13

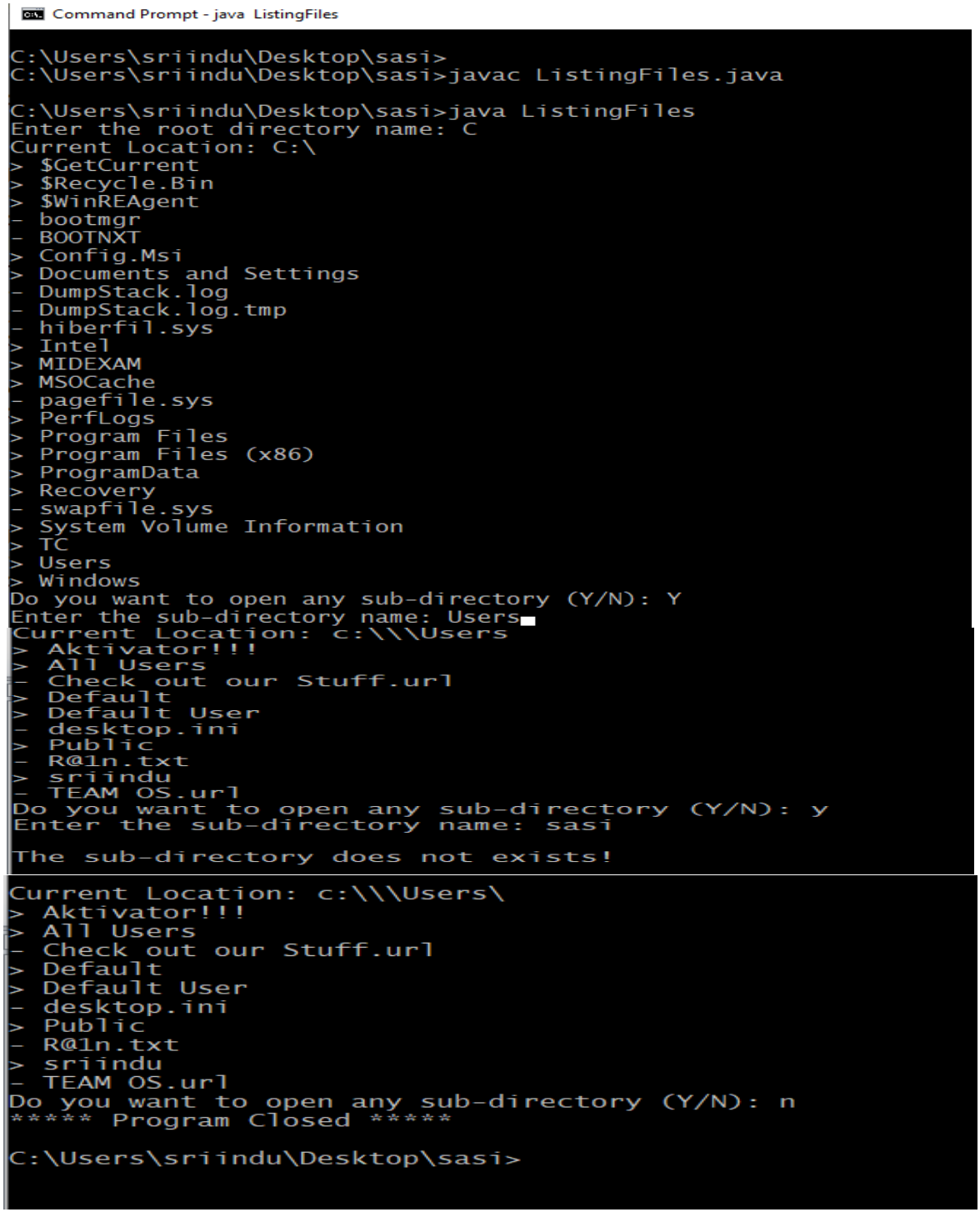
Write a Java program to list all the files in a directory including the files present in all its subdirectories.

Source Code:

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

public class ListingFiles
{
	public static void main(String[] args)
	{

		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 (Y/N):  ");

		     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 exists!");
		            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:



Week 14

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

Source Code:

```
public class QuickSortOnStrings
{
String names[];
int length;

public static void main(String[] args)
{
  QuickSortOnStrings obj = new QuickSortOnStrings();
   String stringsList[] = {"Sasi", "Priya", "Abi", "Savitha", "Hema", "Iniya", "Viji"};
   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:

```
C:\Users\sriindu\Desktop\sasi>javac QuickSortOnStrings.java

C:\Users\sriindu\Desktop\sasi>java QuickSortOnStrings

       Abi Hema Iniya Priya Sasi Savitha Viji
```

Week 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.

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;
		System.out.print("Enter the list size: ");
		size = read.nextInt();

		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);
	}
}
```

Output:

```
C:\Users\sriindu\Desktop\sasi>javac BubbleSort.java

C:\Users\sriindu\Desktop\sasi>java BubbleSort
        Enter the list size: 5
        Enter any 5 integer numbers:
        3
        5
        1
        4
        2

        5       4       3       2       1

        Total number of Interchanges is 4
```