Practical 1

1.) Design a class Complex having a real part (x) and an imaginary part (y). Provide methods to perform the following on complex numbers:

1. Add two complex numbers.
2. Multiply two complex numbers.
3. toString() method to display complex numbers in the form: x + i y

```
import java.util.*;

class Complex {
   int x, y;

   public Complex(int r, int i) {
      this.x = r;
      this.y = i;
   }

   public Complex add(Complex o) {
      return new Complex(
            this.x + o.x,
            this.y + o.y
      );
   }

   public Complex multiply(Complex o) {
      return new Complex(
            this.x * o.x + this.y * o.y,
            this.x * o.y + o.x * this.y
      );
   }


   public String toString() {
      return (x + "+" + y + "i");
   }


   public static void main(String[] args) {
      System.out.println("haa bhyii,ki Haal H!");
      Complex c1 = new Complex(2, 3);
      Complex c2 = new Complex(4, 5);
      System.out.println("complex numbers in the form: x + i y are: ");
      System.out.println(c1.toString());
      System.out.println(c2.toString());
      System.out.println("sum is " + c1.add(c2));
      System.out.println("product is " + c1.multiply(c2));

   }
}
```

PRACTICAL 2

Objective

Create a class TwoDim which contains private members as x and y coordinates in package P1. Define the default constructor, a parameterized constructor and override toString()

method to display the co-ordinates. Now reuse this class and in package P2 create another class ThreeDim, adding a new dimension as z as its private member. Define the constructors for the subclass and override toString() method in the subclass also. Write appropriate methods to show dynamic method dispatch. The main() function should be in a package P.

```
/**** P1/TwoDim.java ****/
package P1;
public class TwoDim {
 private int x;
 private int y;
 public TwoDim() {
 this.x = 0;
 this.y = 0;
 }
 public TwoDim(int x, int y) {
 this.x = x;
 this.y = y;
 }
 @Override
 public String toString() {
 return "Coordinate: x = " + x + " y = " + y;
 }
}
/**** P2/ThreeDim.java ****/
package P2;
import P1.*;
public class ThreeDim extends TwoDim {
 private int z;
 public ThreeDim() {
 super(0, 0);
 this.z = 0;
 }
 public ThreeDim(int x, int y, int z) {
 super(x, y);
 this.z = z;
 }
 @Override
 public String toString() {
 return super.toString() + " z = " + z;
 }
}
/**** P/Main.java ****/
package P;
import P1.*;
import P2.*;
public class Main {
 public static void main(String[] args) {
 TwoDim ref;
 ref = new TwoDim(1, 2);
 System.out.println(ref);
 ref = new ThreeDim(3, 4, 5);
 System.out.println(ref);
 }
}
```

PRACTICAL 3

Objective

Define an abstract class Shape in package P1. Inherit two more classes: Rectangle in package P2 and Circle in package P3. Write a program to ask the user for the type of shape and then using the concept of dynamic method dispatch, display the area of the appropriate subclass. Also write appropriate methods to read the data. The main() function should not be in any package.

```
/**** P1/Shape.java ****/
package P1;
public abstract class Shape {
 protected abstract void getData() throws java.io.IOException;
 public abstract double area() throws java.io.IOException;
}
/**** P2/Rectangle.java ****/
package P2;
import java.io.*;
import P1.*;
public class Rectangle extends Shape {
 private double length;
 private double breadth;
 protected void getData() throws IOException {
 BufferedReader br = new BufferedReader(new InputStreamReader(
System.in
));
 System.out.print("Enter Length of Rectangle: ");
 length = Double.parseDouble(br.readLine());
 System.out.print("Enter Breadth of Rectangle: ");
 breadth = Double.parseDouble(br.readLine());
 }
 public double area() throws IOException {
 getData();
 return length * breadth;
 }
}
/**** P3/Circle.java ****/
package P3;
import java.io.*;
import P1.*;
public class Circle extends Shape {
 private double radius;
 protected void getData() throws IOException {
 BufferedReader br = new BufferedReader(new InputStreamReader(
System.in
));
 System.out.print("Enter Radius of Circle: ");
 radius = Double.parseDouble(br.readLine());
 }
 public double area() throws IOException {
 getData();
 return Math.PI * radius * radius;
 }
}
/**** Main.java ****/
```

```
import java.io.*;
import P1.*;
import P2.*;
import P3.*;
public class Main {
 static int getShapeType() throws IOException {
 BufferedReader br = new BufferedReader(new InputStreamReader(
System.in
));
 System.out.println("==============\n SHAPE TYPE \n==============");
 System.out.println(" (1) Rectangle\n (2) Circle");
 System.out.print("Enter Choice: ");
 return Integer.parseInt(br.readLine());
 }
 public static void main(String[] args) throws IOException {
 Shape ref;
 boolean flag = false;
 while (!flag) {
 switch (getShapeType()) {
 case 1:
 flag = true;
 ref = new Rectangle();
 System.out.println("Area: " + ref.area() + " sq units");
 break;
 case 2:
 flag = true;
 ref = new Circle();
 System.out.println("Area: " + ref.area() + " sq units");
 break;
 default:
 System.err.println("Invalid Option");
 break;
 }
 }
 }
}
-----------------------------------------------------------
```

PRACTICAL 4

Objective

Create an Exception subclass UnderAge, which prints “Under Age” along with the age value when an object of UnderAge class is printed in the catch statement. Write a class exceptionDemo in which the method test() throws UnderAge exception if the variable age passed to it as argument is less than 18. Write main() method also to show working of the program.

```
/**** UnderAge.java ****/
public class UnderAge extends Exception {
 final private int age;
 public UnderAge(int age) {
 this.age = age;
 }
 @Override
 public String getMessage() {
 return "UnderAge: " + age + " is less than 18";
 }
}
/**** exceptionDemo.java ****/
import java.util.Scanner;
class exceptionDemo {
 static void test(int age) throws UnderAge {
 if (age < 18)
 throw new UnderAge(age);
 }
 public static void main(String[] args) {
 Scanner sc = new Scanner(System.in);
 System.out.print("Enter Age: ");
 int age = sc.nextInt();
 try {
 test(age);
 System.out.println("Test Successful");
 } catch (UnderAge e) {
 System.err.println(e.getMessage());
 System.out.println("Test Unsuccessful");
 } finally {
 sc.close();
 }
 }
}
------------------------------------------------------------------------
```

PRACTICAL 5

Objective

Write a program to implement stack. Use exception handling to manage underflow and overflow conditions.

```
/**** Stack.java ****/
public class Stack {
 // Top of the Stack
 private int tos;
 // Array of Elements
 private int[] array;
 // Size of the Stack
 final private int size;
 /**
 * Public Constructor for Stack Objects
 *
 * @param size Size of the Stack
 */
 public Stack(int size) {
 this.tos = -1;
 this.size = size;
 this.array = new int[this.size];
 }
 /**
 * Push an element to the top of the stack
 *
 * @param e Element to be pushed
 * @throws StackException Stack Overflow
 */
 public void push(int e) throws StackException {
 if (tos == size - 1)
 throw new StackException("Stack Overflow: could not push " + e);
 else
 this.array[++this.tos] = e;
 }
 /**
 * Pop an element from the stack
 *
 * @return Object on top of the stack
 * @throws StackException Stack Underflow
 */
 public int pop() throws StackException {
 if (this.tos < 0) {
 throw new StackException("Stack Underflow:could not pop");
 } else
 return this.array[this.tos--];
 }
/**
 * Index of the element at the top of the stack
 *
 * @return Index of Generic Element
 */
 public int getTOS() {
 return this.tos;
 }
 /**
 * Representation of Stack Object
 *
 * @return String Representation
 */
 @Override
 public String toString() {
 return "Stack<size=" + this.size + ">";
 }
}
/**** StackException.java ****/
public class StackException extends Exception {
 final private String message;
 public StackException(String message) {
 this.message = message;
 }
 @Override
 public String getMessage() {
 return this.message;
 }
}
/**** Main.java ****/
import java.util.Random;
public class Main {
 public static void main(String[] args) {
 int r;
 Stack stack = new Stack(10);
 Random random = new Random(1337);
 System.out.println("Created stack of size 10...");
 System.out.println("Pushing integers onto stack...");
 while (true) {
```

```
 r = random.nextInt(100);
 System.out.println("Pushing " + r + "...");
 try {
 stack.push(r);
 System.out.println(
"Elements in Stack = " + (stack.getTOS() +1)
);
 } catch (StackException e) {
 System.err.println(e.getMessage());
 break;
 }
 }
 System.out.println("Popping integers from stack...");
 while (true) {
 System.out.println(
"Elements in Stack = " + (stack.getTOS() +1)
);
 try {
 System.out.println("Popped " + stack.pop() + "...");
 } catch (StackException e) {
 System.err.println(e.getMessage());
 break;
 }
 }
 }
}
-----------------------------------------------------------------------
```

PRACTICAL 6

Objective

Write a program that copies content of one file to another. Pass the names of the files through command-line arguments.

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


public class CopyFromFileaToFileb {
    public static void copyContent(File a, File b)
    throws Exception
    {

        FileInputStream in =new FileInputStream(a);
        FileOutputStream out =new FileOutputStream(b);

        try{
            int n;
            while((n= in.read()) != -1){
                out.write(n);
            }
        }
        finally {
            if (in!=null){
                in.close();
            }
            if (in!=null){
                out.close();
            }
        }
        System.out.println("file copied");
    }

    public static void main (String[] args) throws Exception{
        Scanner sc = new Scanner(System.in);
        System.out.println("enter source file name");
        String a = sc.nextLine();
        File x= new File(a);

        System.out.println("enter destination file name");
        String b = sc.nextLine();
        File y = new File(b);

        copyContent(x,y);
    }


}
--------------------------------------------------------------------------------
```

Practical 7
: Write a program to read a file and display
only those lines that have the first two characters as
'//' (Use try with resources).

```
import java.io.*;
class TryWith
{
public static void main(String args[])throws IOException
{
File infile=new File("f1.txt");
BufferedReader br=new BufferedReader(new FileReader(infile));
try(br)
{
String str,s;
while((str=br.readLine())!=null)
{
str=str.trim();
if(str.substring(0,2).equals("//"))
{
System.out.println(str.substring(2));
}
else
continue;
}
}
finally
{
br.close();
}
}
}
```

Practical 8

Write a program to handle mouse events(Clicked, Entered, Exited, Presses, and Released).

```
import java.awt.*;
import java.awt.event.*;
class MouseDemo extends Frame
{
private Label l = new Label();
MouseDemo()
{
setSize(300, 200);
setLocation(0, 0);
setTitle("Mouse Tester");
setVisible(true);
setLayout(new FlowLayout());
add(l);
addComponentListener(new ComponentAdapter()
{
public void componentResized(ComponentEvent ce)
{
addMouseListener(new MouseAdapter()
{
public void mouseEntered(MouseEvent e)
{
l.setText("Mouse entered");
setSize(200, 200);
}
public void mouseClicked(MouseEvent e)
{
l.setText("Mouse Clicked");
setSize(400, 400);
}
public void mousePressed(MouseEvent e)
{
l.setText("Mouse Pressed");
setSize(250, 250);
}
public void mouseReleased(MouseEvent e)
{
l.setText("Mouse Released");
setSize(350, 300);
}
/*public void mouseExited(MouseEvent e)
{
l.setText("Mouse Exited.");
}*/
});
}
});
}
public static void main(String args[])
{
```

```
new MouseDemo();
}
```

Practical 9
Using AWT, write a program to display a string in frame
window with pink colour as background.

Code:-

```
import java.awt.*;
import java.awt.event.*;
public class PinkFrame extends Frame {
Label l;
PinkFrame() {
l = new Label("This is a Label");
l.setBounds(25, 50, 250, 30);
l.setAlignment(Label.CENTER);
this.setBackground(Color.PINK);
this.add(l);
this.setSize(300, 100);
this.setLayout(null);
this.setVisible(true);
this.addWindowListener(new WindowAdapter() {
public void windowClosing(WindowEvent e) {
dispose();
}
});
}
public static void main(String[] args) {
new PinkFrame();
}
}
```

Practical 10
Using AWT, write a program to create two buttons
named “Red” and “Blue”. When a button is pressed the
background color should be set to the color named by the
button’s label.

Code:-

```
import java.awt.*;
import java.awt.event.*;
public class ButtonDemo extends Frame implements ActionListener {
Button btnRed, btnBlue;
ButtonDemo() {
btnRed = new Button("Red");
btnRed.setBounds(25, 50, 250, 30);
btnRed.addActionListener(this);
this.add(btnRed);
btnBlue = new Button("Blue");
btnBlue.setBounds(25, 100, 250, 30);
btnBlue.addActionListener(this);
this.add(btnBlue);
this.setSize(300, 160);
this.setLayout(null);
this.setVisible(true);
this.addWindowListener(new WindowAdapter() {
public void windowClosing(WindowEvent e) {
dispose();
}
});
}
public static void main(String[] args) {
new ButtonDemo();
}
public void actionPerformed(ActionEvent e) {
if (e.getSource() == btnRed) {
this.setBackground(Color.RED);
}
else if (e.getSource() == btnBlue) {
this.setBackground(Color.BLUE);
}
}
}
```

Practical 11

Using AWT, write a program which responds to KEY_TYPED event and updates the status window with message ("Typed character is: X"). Use adapter class for other two events.

Code :-

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
class KbdAdapter extends KeyAdapter {
Label l;
KbdAdapter(Label l) {
this.l = l;
}
public void keyTyped(KeyEvent e) {
l.setText("Typed character is: " + e.getKeyChar());
}
public void keyPressed(KeyEvent e) {
System.out.println("Pressed character is: " + e.getKeyChar());
}
public void keyReleased(KeyEvent e) {
System.out.println("Released character is: " + e.getKeyChar());
}
}
public class Key extends Frame {
Label l;
Key() {
Font myFont1 = new Font("Serif", Font.PLAIN, 20);
l = new Label("Start");
l.setBounds(25, 50, 250, 30);
l.setFont(myFont1);
l.setAlignment(Label.CENTER);
this.addKeyListener(new KbdAdapter(l));
this.add(l);
this.setSize(300, 110);
this.setLayout(null);Q
this.setVisible(true);
this.addWindowListener(new WindowAdapter() {
public void windowClosing(WindowEvent e) {
dispose();
}
});
}
public static void main(String[] args) {
new Key();
}
}
```

Practical 12
Using AWT, write a program to create two buttons labelled ‘A’ and ‘B’. When button ‘A’ is pressed, it displays your personal information (Name, Course, Roll No, College) and when button ‘B’ is pressed, it displays your CGPA in previous semester.
Code :-

```
import java.awt.*;
import java.awt.event.*;
public class Info extends Frame implements ActionListener {
Button btnInfo, btnCGPA;
Info() {
super("Student Details");
// Button A
btnInfo = new Button("A");
btnInfo.setBounds(25, 125, 450, 100);
btnInfo.addActionListener(this);
this.add(btnInfo);
// Button B
btnCGPA = new Button("B");
btnCGPA.setBounds(25, 300, 450, 100);
btnCGPA.addActionListener(this);
this.add(btnCGPA);
this.setSize(500, 500);
this.setLayout(null);
this.setVisible(true);
this.setLocationRelativeTo(null);
this.addWindowListener(new WindowAdapter() {
public void windowClosing(WindowEvent e) {
dispose();
}
});
}
public void actionPerformed(ActionEvent e) {
if(e.getSource() == btnInfo) {
new PersonalInfo("Nikhil Bainsla","BSc(Hons) Computer
Science","20211439","Ramanujan college");
}else{
new CGPA("9.5");
}
}
}
```

```
public static void main(String[] args) {
new Info();
}
}
```