Government Engineering College
Sec-28 Gandhinagar

Sem: - IV

Subject: - Object Oriented Programming -I

Subject Code: - 3140705

Government Engineering College

Sec-28 Gandhinagar

Certificate

This is to certify that

Mr./Ms. …….. Of class

…….Division ……., Enrollment No. …….. Has

Satisfactorily completed his/her term work in

…….. Subject for the term ending in

……2022.

Date: -

Contents

Institute Vision/Mission

Vision:

- To be a premier engineering institution, imparting quality education for innovative solutions relevant to society and environment.

Mission:

- To develop human potential to its fullest extent so that intellectual and innovative engineers can emerge in a wide range of professions.
- To advance knowledge and educate students in engineering and other areas of scholarship that will best serve the nation and the world in future.
- To produce quality engineers, entrepreneurs and leaders to meet the present and future needs of society as well as environment.

Computer Engineering Department Vision/Mission

Vision:

- To achieve excellence for providing value based education in Computer Engineering through innovation, team work and ethical practices.

Mission:

- To produce computer science and engineering graduates according to the needs of industry, government, society and scientific community.
- To develop partnership with industries, government agencies and R and D Organizations
- To motivate students/graduates to be entrepreneurs.
- To motivate students to participate in reputed conferences, workshops, symposiums, seminars and related technical activities

Program Educational Outcome (PEO)

- To provide students with a strong foundation in the mathematical, scientific and engineering fundamentals necessary to formulate, solve and analyze engineering problems and to prepare them for graduate studies, R&D, consultancy and higher learning.
- To develop an ability to analyze the requirements of the software, understand the technical specifications, design and provide novel engineering solutions and efficient product designs.
- To provide exposure to emerging cutting edge technologies, adequate training & opportunities to work as teams on multidisciplinary projects with effective communication skills and leadership qualities.
- To prepare the students for a successful career and work with values & social concern bridging the digital divide and meeting the requirements of Indian and multinational companies.
- To promote student awareness on the life-long learning and to introduce them to professional ethics and codes of professional practice

PSO

By the completion of Computer Engineering program the student will have following Program specific outcomes.

- Design, develop, test and evaluate computer-based systems by applying standard software engineering practices and strategies in the area of algorithms, web design, data structure, and computer network
- Apply knowledge of ethical principles required to work in a team as well as to lead a team

POs

Engineering Graduates will be able to:

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

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.

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, and environmental considerations.

4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering

activities with an understanding of the limitations.

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 relevant to the professional engineering practice.

7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

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 and leader in a team, to manage projects and in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Object Oriented Programming -I (3140705)

Course Outcomes (COs)

CO-1	
CO-2	
CO-3	
CO-4	
CO-5	

Assignment Index

Sr. No	Assignment	Date	Page No.	Sign
1	Assignment 1			
2	Assignment 2			
3	Assignment 3			
4	Assignment 4			

Practical Index

Sr. No	Assignment	Date	Page No.	Sign
1	Practical 1	17/01/2022		
2	Practical 2	17/01/2022		
3	Practical 3	24/01/2022		
4	Practical 4	24/01/2022		
5	Practical 5	31/01/2022		
6	Practical 6	31/01/2022		
7	Practical 7	07/02/2022		
8	Practical 8	07/02/2022		
9	Practical 9	14/02/2022		
10	Practical 10	14/02/2022		
11	Practical 11	21/02/2022		
12	Practical 12	21/02/2022		
13	Practical 13	28/02/2022		
14	Practical 14	07/03/2022		
15	Practical 15	14/03/2022		
16	Practical 16	21/03/2022		
17	Practical 17	28/03/2022		
18	Practical 18	04/04/2022		

19	Practical 19	11/04/2022		
20	Practical 20	18/04/2022		
21	Practical 21	02/05/2022		
22	Practical 22	09/05/2022		
23	Practical 23	16/05/2022		
24	Practical 24	23/05/2022		
25	Practical 25	30/05/2022		

ASSIGNMENT – 1

a) Write a program that reads a number in meters, converts it to feet, and displays the result using command line argument.

SOLUTION:

```
class Assign1_a {
   public static void main(String[] args) {
      System.out.println("Enter the value of meter:");
      float a = Integer.parseInt(args[0]);
      float ans = a * 3.28084f;
      System.out.println("Answer is"+ ans+" feet");
   }
}
```

OUTPUT:

```
C:\Users\Hp>cd/

C:\>cd OOP Practicles

C:\OOP Practicles>javac Assign1_a.java

C:\OOP Practicles>java Assign1_a 17
Enter the value of meter:
Answer is55.774277 feet
```

b) Write a test program that prompts the user to enter numbers, invoke a method to reverse the numbers, display the numbers.

SOLUTION:

```
import java.util.Scanner;
class Assign1_b {
   public static void main(String[] args) {
      Scanner sc=new Scanner(System.in);
      int a[]=new int[10];
      System.out.println("Enter the 10 Integer.....");
      for (int i=0;i<a.length;i++){
         a[i]=sc.nextInt();
```

```
        }
        for (int i=0;i<(a.length)-1;i++){
            for (int j=0;j<(a.length)-i-1;j++){
                int temp;
                temp=a[j];
                a[j]=a[j+1];
                a[j+1]=temp;
            }
        }
        System.out.println("Reversing number.....");
        for (int i=0;i<(a.length);++i){
            System.out.print(a[i]);
            System.out.print(" ");
        }
    }
}
```

OUTPUT:

ASSIGNMENT – 2

a) Write a program to demonstrate the user of array of Objects and static variable

SOLUTION:

```
class Demo{
   int rollNo;
    String name;
    static String name1;
   public Demo(int rollNo, String name,String name1) {
      this.rollNo = rollNo;
      this.name=name;
      this.name1=name1;
   }
      public void display(){
         System.out.println(rollNo+"\t"+name);
      }
   }
class Assign2_a {
   public static void main(String[] args) {
      Demo []a =new Demo[8];

      a[0]=new Demo(12,"Manav","GEC GANDHINAGAR");
      a[1]=new Demo(15,"Swayam","GEC GANDHINAGAR");
      a[2]=new Demo(16,"Dhoda Dhruv","GEC GANDHINAGAR");
      a[3]=new Demo(25,"Raj","GEC GANDHINAGAR");
      a[4]=new Demo(28,"Rahul","GEC GANDHINAGAR");
      a[5]=new Demo(38,"Harshil","GEC GANDHINAGAR");
      a[6]=new Demo(42,"Manav","GEC GANDHINAGAR");
      a[7]=new Demo(65,"Dhruv","GEC GANDHINAGAR");
```

```
        System.out.println("ROLLNO\tNAME\tcollage");
        System.out.println("------------------");
        for (int i=0;i<a.length;i++){
            System.out.println(a[i].rollNo+"     "+a[i].name+"    "+a[i].name1);
        }
    }
}
```

OUTPUT:

```
C:\OOP Practicles>javac Assign2_a.java

C:\OOP Practicles>java Assign2_a
ROLLNO  NAME    collage
------------------
12      Manav    GEC GANDHINAGAR
15      Swayam    GEC GANDHINAGAR
16      Dhoda Dhruv    GEC GANDHINAGAR
25      Raj    GEC GANDHINAGAR
28      Rahul    GEC GANDHINAGAR
38      Harshil    GEC GANDHINAGAR
42      Manav    GEC GANDHINAGAR
65      Dhruv    GEC GANDHINAGAR
```

b) Create a class Account with two overloaded constructors. The first constructor is used for initializing, the name of account holder, the account number and the initial amount in the account. The second constructor is used for initializing the name of the account and the current balance. The Account class is having method Deposite(), Withdraw(), and Get_Balance().Make the necessary assumption for data members and return types of methods. Create objects of Account class and use them.

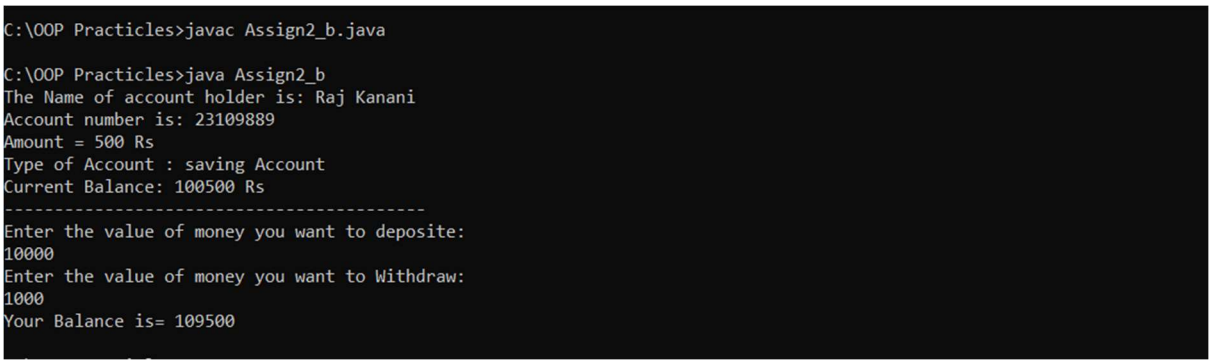
SOLUTION:

```
import java.util.Scanner;
class Account{
    String name_hold,acc_name;
     static int amount,curr_balance;
     int acc_num;
   int a,b,c;
    Scanner sc=new Scanner(System.in);
    Account(String name_hold,int acc_num,int amount){
        this.name_hold=name_hold;
        this.acc_num=acc_num;
        this.amount=amount;
        System.out.println("The Name of account holder is: "+name_hold);
```

```
        System.out.println("Account number is: "+acc_num);
        System.out.println("Amount = "+amount + " Rs");
    }
    Account(String acc_name,int curr_balance){
      this.acc_name=acc_name;
      this.curr_balance=curr_balance;
        System.out.println("Type of Account : " +acc_name);
        System.out.println("Current Balance: "+curr_balance+ " Rs");
    }
    void Deposite(){
        System.out.println("Enter the value of money you want to deposite:");
         a=sc.nextInt();
      c=a+curr_balance;
    }
    void Withdraw(){
        System.out.println("Enter the value of money you want to Withdraw:");
      b= sc.nextInt();
    }
    void Get_balance(){
        if (curr_balance<b){
            System.out.println(".....ERROR.....");
        }
        System.out.println("Your Balance is= "+(c-b));
    }
}
class Assign2_b {
    public static void main(String[] args) {
        Account a1=new Account("Raj Kanani",23109889,500);
        Account a2=new Account("saving Account",100500);
        System.out.println("----------------------------------------");
```

```
    a1.Deposite();
    a1.Withdraw();
    a1.Get_balance();
  }
}
```

OUTPUT:



c) The Airplane class has three subclasses named B747, B757, and B767. Each plane type can transport different no. of passengers. Each airplane object has a unique serial no. Write an application that declares this class hierarchy. Instantiate several types of airplanes and display them. Override the toString() method of object to return a string with the type, serial no. and capacity.

SOLUTION:

```
class Airplane{
   int Serial_no,capacity;
   String air_name;
  Airplane(int s,int ca,String nam){
     Serial_no=s;
     capacity=ca;
     air_name=nam;
  }
  public void tostring(){
     System.out.println("Serial number is: "+Serial_no);
     System.out.println("Airplane name is: "+air_name);
     System.out.println("CAPACITY : "+capacity);
  }
}
```

```
class B747 extends Airplane{
   B747(int s,int ca,String nam){
       super(s,ca,nam);
   }
}
class B757 extends  Airplane{
   B757(int s,int ca,String nam){
       super(s,ca,nam);
   }
}
class B767 extends  Airplane{
   B767(int s,int ca,String nam){
       super(s,ca,nam);
   }
}
class Assign2_c {
    public static void main(String[] args) {
        B747 b1=new B747(1023,500,"Indian Airline");
        B757 b2=new B757(1000,400,"Indigo");
        B767 b3=new B767(2021,700,"Decan Airline");
        System.out.println(".....Details of the Airplan.....");
        System.out.println("--------------------------------");
        b1.tostring();
        System.out.println("--------------------------------");
        b2.tostring();
        System.out.println("--------------------------------");
        b3.tostring();
    }
}
```

OUTPUT:

```
C:\OOP Practicles>javac Assign2_c.java

C:\OOP Practicles>java Assign2_c
...........Details of the Airplan.............
------------------------------
Serial number is: 1023
Airplane name is: Indian Airline
CAPACITY : 500
------------------------------
Serial number is: 1000
Airplane name is: Indigo
CAPACITY : 400
------------------------------
Serial number is: 2021
Airplane name is: Decan Airline
CAPACITY : 700
```

d) Write a program that illustrates interface inheritance. Interface P is extended by P1 and P2. Interface P12 inherited from both p1 and p2. Each interface declares constant and one method. Class Q implements P12. Instantiate Q and invoke each of its method. Each method displays one of the constants.

SOLUTION:

```
interface p{
   int a=52;
   void display();
}
interface p1 extends p{
   int b=78;
   void display1();
}
interface p2 extends p{
   int c=56;
   void display2();
}
interface p12 extends p1,p2{
   int d=67;
   void display3();
}
class Q implements p12{
   @Override
   public void display() {
      System.out.println("This is the interface p and his constant value is :"+a);
```

```
    }
    @Override
    public void display1() {
        System.out.println("This is the interface p1 and his constant value is :"+b);
    }
    @Override
    public void display2() {
        System.out.println("This is the interface p2 and his constant value is :"+c);
    }
    @Override
    public void display3() {
        System.out.println("This is the interface p12 and his constant value is :"+d);
    }
}
class Assign2_d {
    public static void main(String[] args) {
        p e=new Q();
        p1 e1=new Q();
        p2 e2=new Q();
        p12 e3=new Q();
        e3.display();
        e3.display1();
        e3.display2();
        e3.display3();
}
}
```

OUTPUT:

```
C:\OOP Practicles>javac Assign2_d.java

C:\OOP Practicles>java Assign2_d
This is the interface p and his constant value is :52
This is the interface p1 and his constant value is :78
This is the interface p2 and his constant value is :56
This is the interface p12 and his constant value is :67

C:\OOP Practicles>_
```

e) Write a program to implement an Abstract class Shape which contains Abstract method Area(). Create two other classes Circle and Square which overrides the method Area() and find the area of rectangle and square in respective classes. Write demo class

SOLUTION:

```
import java.util.Scanner;
abstract class Shape{
   abstract void Area();
}
class Circle extends Shape{
   Scanner sc=new Scanner(System.in);
     int r;
   @Override
    void  Area() {
       System.out.println("Enter the value of radius: ");
       r=sc.nextInt();
       double result=Math.PI*r*r;
       System.out.println("the area of circle is: "+result);
   }
}
class Square extends Shape{
   Scanner sc=new Scanner(System.in);
   int l;
   @Override
   void Area() {
       System.out.println("Enter the value of length: ");
       l=sc.nextInt();
      int  result=l*l;
```

```
        System.out.println("The area of square is: "+result);
    }
}
class Rectangle extends Shape{
    Scanner sc=new Scanner(System.in);
    int w,h;
    @Override
    void Area() {
        System.out.println("Enter the value of width: ");
        w=sc.nextInt();
        System.out.println("Enter the value of height: ");
        h=sc.nextInt();
        int result=w*h;
        System.out.println("The area of rectangle: "+result);
    }
}
class Assign2_e {
    public static void main(String[] args) {
        Circle c1=new Circle();
        Square s1=new Square();
        Rectangle r1=new Rectangle();
        c1.Area();
        s1.Area();
        r1.Area();
    }
}
```

OUTPUT:

```
C:\OOP Practicles>javac Assign2_e.java

C:\OOP Practicles>java Assign2_e
Enter the value of radius:
10
the area of circle is: 314.1592653589793
Enter the value of length:
150
The area of square is: 22500
Enter the value of width:
200
Enter the value of height:
150
The area of rectangle: 30000

C:\OOP Practicles>
```

f) Write an application that illustrates how a method can invoke a super class method. Class I2 is extended by J2. Class J2 is extended by K2. Each of these class defines a getDescription() method that returns a string. That String includes a description of the class plus descriptions of each super class. Instantiate each object of these classes and invoke the getDescription () method.

SOLUTION:

```
class I2{
   void getDescription(){
      String s="I am a method of class I2.....";
      System.out.println(s);
   }
}
class J2 extends I2{
   String n1;
   @Override
   void getDescription() {
      String s="I am a method of class J2.....";
      super.getDescription();
      System.out.println(s);
   }
}
class K2 extends J2{
   @Override
   void getDescription() {
```

```
        String s="I am a method of class K2.....";
        super.getDescription();
        System.out.println(s);
    }
}

class Assign2_f {
    public static void main(String[] args) {
        K2 obj=new K2();
        obj.getDescription();
    }
}
```

OUTPUT:

```
C:\OOP Practicles>javac Assign2_f.java

C:\OOP Practicles>java Assign2_f
I am a method of class I2.........
I am a method of class J2.......
I am a method of class K2.......

C:\OOP Practicles>
```

ASSIGNMENT - 3

a) Write a Program to demonstrate the use of Multithreading.

SOLUTION:

```
class MyNewThr1 extends Thread {
   public void run() {
      int i=0;
      while (i<4) {
         System.out.println("i am threads");
         i++;
      }
   }
}
class MyNewThr2 extends Thread{
   int i=0;
   public void run(){
      while(i<4){
         System.out.println("Thread2.....");
         i++;
      }
   }
}
class Assign3_a {
   public static void main(String[] args) {
      MyNewThr1 t1=new MyNewThr1();
      MyNewThr2 t2=new MyNewThr2();
      t1.start();
      try{
         t1.join();                    /*this join condition to provide first t1 object execute
                                       after complete t1 then t2 will  execute     */

      }
      catch (Exception e){
         System.out.println(e);
```

```
        }
        t2.start();
    }
}
```

OUTPUT:

```
C:\OOP Practicles>javac Assign3_a.java

C:\OOP Practicles>java Assign3_a
i am threads
i am threads
i am threads
i am threads
Thread2.............
Thread2.............
Thread2.............
Thread2.............

C:\OOP Practicles>
```

b) Write a Program in which the area of room is calculated and the cost of white wash is also evaluated. Further, include the provision for window on any type of the walls. The input regarding the parameters including length, breadth and height of the room are taken through command line. If there is a window, then its parameter including length and breadth are also taken through command line. If these input parameters are below 1, then raise an exception; otherwise calculate the area and cost and display the result. (note that in order to calculate the area of the room to be painted the area of window must be deducted from the total of the room.)

SOLUTION:

```
import java.util.Scanner;
public class Assign3_b {
    public static void main(String[] args) {
        Scanner sc =new Scanner(System.in);


        System.out.println("The rate of white wash is = 15 per m^2");
        System.out.println("The window size = 18 Square feet");
        System.out.println("-----------------------------------------");
        try {
            System.out.println("Enter the length:");
            int length = sc.nextInt();
            System.out.println("Enter the Width:");
            int width = sc.nextInt();
            System.out.println("Enter the Height:");
            int height = sc.nextInt();
            System.out.println("Enter the number of window in the room:");
            int numberOfWindow = sc.nextInt();
```

```
        if(length<1 || width<1 || height<1){
            throw new ArithmeticException();
        }
        int totalSurfaceArea= 2 * ((length * width) + (length * height) + (width * height));
        int areaOfFloor = length * width;
        int areaOfWindow = 18 * numberOfWindow;
        int totalPaintArea = totalSurfaceArea - areaOfFloor - areaOfWindow;
        int totalCost_of_Paint = 15 * totalPaintArea;
        System.out.println("The area of Room = "+ totalSurfaceArea);
        System.out.println("The area of window =" +areaOfWindow);
        System.out.println("Total Paint Area= "+totalPaintArea);
        System.out.println("Total cost of paint = "+totalCost_of_Paint+ " RS" );
      }
      catch(Exception e){
        System.out.println("The number is below 1 please enter valid number!!");
      }
   }
}
```

OUTPUT:

```
C:\OOP Practicles>javac Assign3_b.java

C:\OOP Practicles>java Assign3_b
The rate of white wash is = 15 per m^2
The window size = 18 Square feet
----------------------------------------
Enter the length:
100
Enter the Width:
150
Enter the Height:
200
Enter the number of window in the room:
3
The area of Room = 130000
The area of window =54
Total Paint Area= 114946
Total cost of paint = 1724190 RS

C:\OOP Practicles>
```

ASSIGNMENT - 4

a) Create a class called Student. Write a student manager program to manipulate the student information from files by using FileInputStream and FileOutputStream.

SOLUTION:

```
import java.io.FileInputStream;
import java.io.FileOutputStream;
class Assign4_a {
   public static void main(String[] args) {
      System.out.println("-------Writing Data in File-------");
      try {
         FileOutputStream fout = new FileOutputStream("studentInfo.txt");
         String str = " Name : Raj Kanani\nStream : Computer Engineering,\nSem : 4th Sem\nSub: Object Oriented Programming-I";
         byte b[] = str.getBytes();
         fout.write(b);
         fout.close();
         System.out.println("successful write.");
      } catch (Exception e) {
         System.out.println(e);
      }
      System.out.println("-------Display Data From File-------");
      try {
         FileInputStream fin = new FileInputStream("studentInfo.txt");
         int i = 0;
         while ((i = fin.read()) != -1) {
            System.out.print((char) i);
         }
         fin.close();
      } catch (Exception e) {
         System.out.println(e);
      }
   }
}
```

OUTPUT:

```
C:\OOP Practicles>javac Assign4_a.java

C:\OOP Practicles>java Assign4_a
-------Writing Data in File-------
successful write.
-------Display Data From File-------
 Name : Raj Kanani
Stream : Computer Engineering,
Sem : 4th Sem
Sub: Object Oriented Programming-I
C:\OOP Practicles>
```

b) Write a GUI program that use button to move the message to the left and right and use the radio button to change the color for the message displayed.

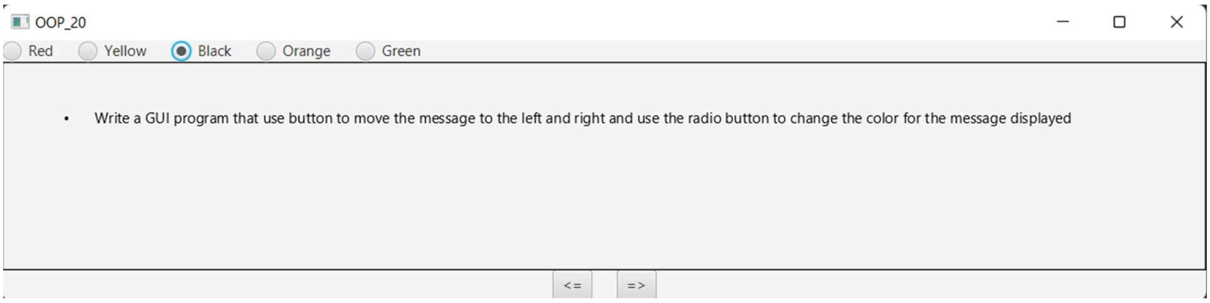
SOLUTION:

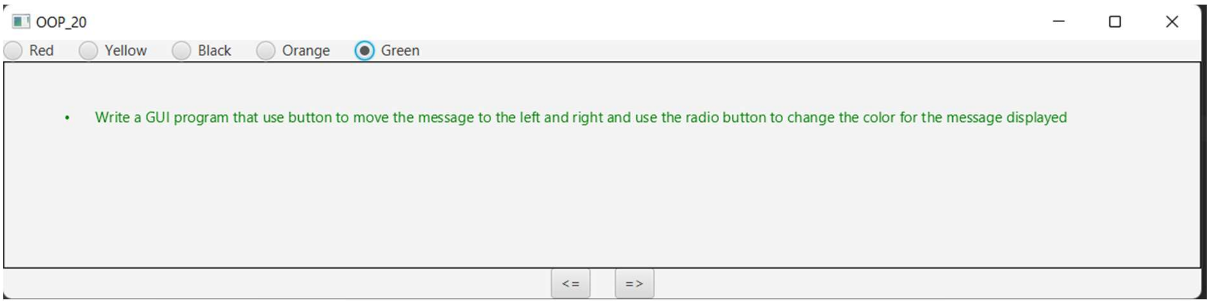
```
import javafx.application.Application;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.geometry.Pos;
import javafx.scene.control.Button;
import javafx.scene.layout.HBox;
import javafx.scene.layout.Pane;
import javafx.scene.layout.BorderPane;
import javafx.scene.text.Text;
import javafx.scene.control.RadioButton;
import javafx.scene.control.ToggleGroup;
import javafx.scene.paint.Color;

public class Assign4_b extends Application {
   protected Text text = new Text(50, 50, "•\tWrite a GUI program that use button to move the message to the left and right and use the radio button to change the color for the message displayed");

   public static void main(String[] args) {
      launch(args);
   }

   @Override
   public void start(Stage primaryStage) {
      HBox paneForButtons = new HBox(20);
      Button btLeft = new Button("<=");
      Button btRight = new Button("=>");
      paneForButtons.getChildren().addAll(btLeft, btRight);
      paneForButtons.setAlignment(Pos.CENTER);
      BorderPane pane = new BorderPane();
      pane.setBottom(paneForButtons);

      HBox paneForRadioButtons = new HBox(20);
      RadioButton rbRed = new RadioButton("Red");
      RadioButton rbYellow = new RadioButton("Yellow");
      RadioButton rbBlack = new RadioButton("Black");
      RadioButton rbOrange = new RadioButton("Orange");
      RadioButton rbGreen = new RadioButton("Green");
      paneForRadioButtons.getChildren().addAll(rbRed, rbYellow,
            rbBlack, rbOrange, rbGreen);

      ToggleGroup group = new ToggleGroup();
      rbRed.setToggleGroup(group);
      rbYellow.setToggleGroup(group);
      rbBlack.setToggleGroup(group);
      rbOrange.setToggleGroup(group);
      rbGreen.setToggleGroup(group);

      Pane paneForText = new Pane();
      paneForText.setStyle("-fx-border-color: black");
      paneForText.getChildren().add(text);
      pane.setCenter(paneForText);
      pane.setTop(paneForRadioButtons);

      btLeft.setOnAction(e -> text.setX(text.getX() - 10));
      btRight.setOnAction(e -> text.setX(text.getX() + 10));
```

```
        rbRed.setOnAction(e -> {
            if (rbRed.isSelected()) {
                text.setFill(Color.RED);
            }
        });

        rbYellow.setOnAction(e -> {
            if (rbYellow.isSelected()) {
                text.setFill(Color.YELLOW);
            }
        });

        rbBlack.setOnAction(e -> {
            if (rbBlack.isSelected()) {
                text.setFill(Color.BLACK);
            }
        });

        rbOrange.setOnAction(e -> {
            if (rbOrange.isSelected()) {
                text.setFill(Color.ORANGE);
            }
        });

        rbGreen.setOnAction(e -> {
            if (rbGreen.isSelected()) {
                text.setFill(Color.GREEN);
            }
        });

        Scene scene = new Scene(pane, 450, 150);
        primaryStage.setTitle("OOP_20");
        primaryStage.setScene(scene);
        primaryStage.show();
    }
}
```

OUTPUT:


OOP_20
Red
Yellow
Black
Orange
Green
Write a GUI program that use button to move the message to the left and right and use the radio button to change the color for the message displayed
<=
=>


OOP_20
Red
Yellow
Black
Orange
Green
Write a GUI program that use button to move the message to the left and right and use the radio button to change the color for the message displayed
<=
=>


OOP_20
Red
Yellow
Black
Orange
Green
Write a GUI program that use button to move the message to the left and right and use the radio button to change the color for the message displayed
<=
=>

PRACTICAL – 1

- Aim: Write a Program that displays Welcome to Java, Learning Java Now and Programming is fun.
- Solution:

```
class Practical_1 {
public static void main(String[] args) {
System.out.println("Welcom to Java");
System.out.println("Learing Java Now");
System.out.println("Programming is Fun");
}
}
```

- Output:

```
M:\java_practical>javac practical1.java

M:\java_practical>java practical1
Welcome To Java
Learning Java Now and Programming is Fun.
```

PRACTICAL – 2

- Aim: Write a program that solves the following equation and displays the value x and y:
  1) 3.4x+50.2y=44.5 2) 2.1x+.55y=5.9 (Assume Cramer's rule to solve equation ax+by=e x=ed-bf/ad-bc cx+dy=f y=af-ec/ad-bc).
- Solution:

```
import java.util.Scanner;
class Practical_2 {
public static void main(String[] args) {
double a , b , e;
double c ,d ,f;
Scanner sc = new Scanner(System.in);
System.out.println("The First Equation is:- ");
System.out.println("aX + bY = e");
System.out.println("Enter the Value of a:- ");
a = sc.nextDouble();
System.out.println("Enter the Value of b:- ");
b = sc.nextDouble();
System.out.println("Enter the Value of e:- ");
e = sc.nextDouble();
System.out.println("The Second Equation is:- ");
System.out.println("cX + dY = f");
System.out.println("Enter the Value of c:- ");
c = sc.nextDouble();
System.out.println("Enter the Value of d:- ");
d = sc.nextDouble();
```

```
System.out.println("Enter the Value of f:- ");
f = sc.nextDouble();
double X = ((e*d)-(b*f))/((a*d)-(b*c));
double Y = ((a*f)-(e*c))/((a*d)-(b*c));
System.out.println("The Value of X is " + X + "The Value of Y is " + Y);
}
}
```

- Output:

```
M:\java_practical>javac practical2.java

M:\java_practical>java practical2
x=2.6239014968614l9
y=0.70873973925639 78
```

PRACTICAL – 3

- Aim: Write a program that reads a number in meters, converts it to feet, and displays the result.
- Solution:

```
import java.util.Scanner;
public class practical_3
{
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
double meter;
System.out.println("Enter the Any Value in Meter:- ");
meter = sc.nextDouble();
double Feet = (meter * 3.28084) ;
System.out.println("Meter = " + meter + " Feet = " +Feet);
}
}
```

- Output:

```
M:\java_practical>javac practical3.java

M:\java_practical>java practical3
Enter the number in meters :
10
The number in feet : 32.804
```

PRACTICAL – 4

- Aim: Body Mass Index (BMI) is a measure of health on weight. It can be calculated by taking your weight in kilograms and dividing by the square of your height in meters. Write a program that prompts the user to enter a weight in pounds and height in inches and displays the BMI.
- Solution:

```
import java.util.Scanner;
public class practical_4 {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
System.out.print("Enter Your weight in Pound :");
double pound = sc.nextDouble();
System.out.print("Enter Your Height in Inch :");
double inch = sc.nextDouble();
double BMI = (pound * 0.45359237) / ((inch * 0.0254)*(inch * 0.0254));
System.out.print("BMI = "+BMI);
}
}
```

- Output:

```
M:\java_practical>java practical4
Enter weight in pound :
5
Enter height in inch :
10
BMI :8.928983661417323
```

PRACTICAL – 5

- Aim: Write a program that prompts the user to enter three integers and display the integers in decreasing order.
- Solution:

```
import java.util.Scanner;
class practical_5 {
public static void main(String[] args) {
int temp;
Scanner sc = new Scanner(System.in);
System.out.print("Enter 1st Integer :");
int a = sc.nextInt();
System.out.print("Enter 2nd Integer :");
int b = sc.nextInt();
if (a < b) {
```

```
temp = a;
a = b;
b = temp;
}
System.out.print("Enter 3rd Integer :");
int c = sc.nextInt();
if (c > b) {
if (c > a) {
temp = c;
c = b;
b = a;
a = temp;
} else {
temp = c;
c = b;
b = temp;
}
}
System.out.print("Decreasing Order :" + a + " " + b + " " + c);
}
}
```

- Output:

```
M:\java_practical>javac practical5.java

M:\java_practical>java practical5
Enter the first integer number :
5
Enter the second integer number :
9
Enter the third integer number :
2
9 5 2
```

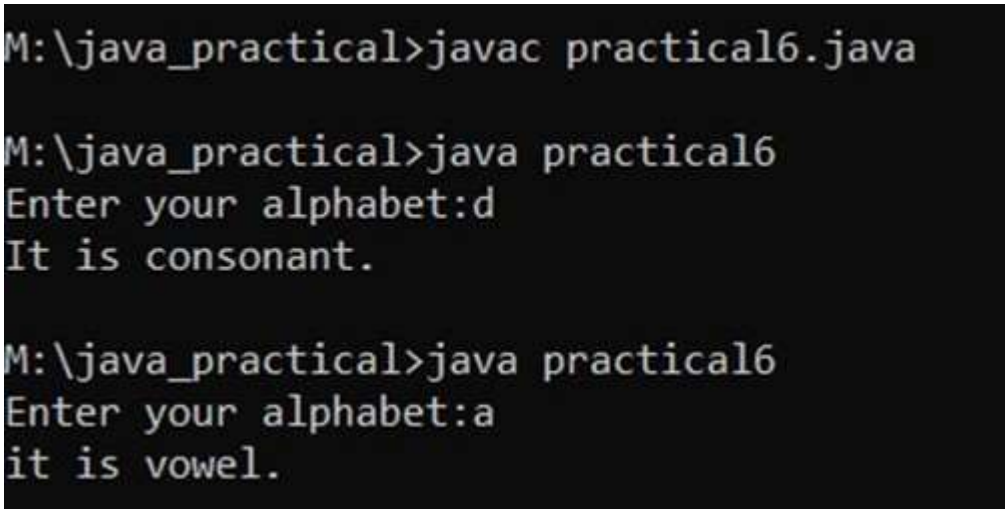
PRACTICAL – 6

- Aim: Write a program that prompts the user to enter a letter and check whether a letter is a vowel or constant.
- Solution:

```
import java.util.Scanner;
class practical_6 {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
System.out.print("Enter Character : ");
char ch = sc.next().charAt(0);
if (ch>=48 && ch<57) {
System.out.println("Enter the Alphabate not Number :)");
}
else{
switch (Character.toLowerCase(ch)) {
```

```
case 'a':
case 'e':
case 'i':
case 'o':
case 'u':
System.out.print(ch + " is vowel");
break;
default:
System.out.print(ch + " is constant");
}}}}
```

- Output:



PRACTICAL – 7

- Aim: Assume a vehicle plate number consists of three uppercase letters followed by four digits. Write a program to generate a plate number.
- Solution:

```
public class practical_7 {
public static void main(String[] args) {
int alpha1 = 'A' + (int) (Math.random() * ('Z' - 'A'));
int alpha2 = 'A' + (int) (Math.random() * ('Z' - 'A'));
int alpha3 = 'A' + (int) (Math.random() * ('Z' - 'A'));
int digit1 = (int) (Math.random() * 10);
int digit2 = (int) (Math.random() * 10);
int digit3 = (int) (Math.random() * 10);
int digit4 = (int) (Math.random() * 10);
System.out.println("" + (char) (alpha1) + ((char) (alpha2)) +
((char) (alpha3)) + digit1 + digit2 + digit3 + digit4);
}
}
```

- Output:

```
M:\java_practical>javac practical7.java

M:\java_practical>java practical7
Plate No. :OGK0146
```

PRACTICAL – 8

- Aim: Write a program that reads an integer and displays all its smallest factors in increasing order. For example if input number is 120, the output should be as follows:2,2,2,3,5.
- Solution:

```
import java.util.Scanner;
public class practical_8 {
public static void main(String[] args) {
int div = 2;
Scanner sc = new Scanner(System.in);
System.out.print("Enter Integer Value : ");
int number = sc.nextInt();
System.out.print("The All Factors of " + number + " is : ");
while (number > 1) {
if (number % div == 0) {
System.out.print(div + ",");
number = number / div;
} else {
div++;
}
}
}
}
```

- Output:

```
M:\java_practical>javac practical8.java

M:\java_practical>java practical8
Enter the no. :
120
2 2 2 3 5
```

PRACTICAL – 9

- Aim: Write a method with following method header.public static int gcd(int num1, int num2) Write a program that prompts the user to enter two integers and compute the gcd of tw integers.
- Solution:

```
import java.util.Scanner;
class practical_9 {
public static int gcd(int num1, int num2) {
```

```
while (num1 != num2) {
if (num1 > num2) {
num1 = num1 - num2;
} else {
num2 = num2 - num1;
}
}
return num1;
}
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
System.out.print("Enter First Number : ");
int number1 = sc.nextInt();
System.out.print("Enter Second Number : ");
int number2 = sc.nextInt();
System.out.print("GCD of " + number1 + " and " + number2 + " = " +
gcd(number1, number2));
}
}
```

- Output:

```
M:\java_practical>javac practical9.java

M:\java_practical>java practical9
Enter the first no. :
12

Enter the second no. :
6

GCD of 12 and 6 is :6
```

PRACTICAL – 10

- Aim: Write a test program that prompts the user to enter ten numbers, invoke a method to reverse the numbers, display the numbers.
- Solution:

```
import java.util.Scanner;
class practical_10 {
public static void reverse(int num[]) {
int j = 0, temp;
while (j <= num.length / 2) {
temp = num[j];
num[j] = num[num.length - 1 - j];
num[num.length - 1 - j] = temp;
j++;
}
}
public static void main(String[] args) {
```

```
int i = 0;
int num_array[] = new int[10];
Scanner sc = new Scanner(System.in);
System.out.println("Enter the Numbers:- ");
for (i = 0; i < 10; i++) {
System.out.print("Number " + (i + 1) + " : ");
num_array[i] = sc.nextInt();
}
reverse(num_array);
System.out.println("After reversing Number in an Array :");
for (i = 0; i < 10 ; i++) {
System.out.println("Number " + (i + 1) + " : " + num_array[i]);
}
}
}
```

- Output:

```
M:\java_practical>javac practical10.java

M:\java_practical>java practical10
Enter the 1 no. :1
Enter the 2 no. :2
Enter the 3 no. :3
Enter the 4 no. :4
Enter the 5 no. :5
Enter the 6 no. :6
Enter the 7 no. :7
Enter the 8 no. :8
Enter the 9 no. :9
Enter the 10 no. :10

After reversing .......

The 10 no. is :10
The 9 no. is :9
The 8 no. is :8
The 7 no. is :7
The 6 no. is :6
The 5 no. is :5
The 4 no. is :4
The 3 no. is :3
The 2 no. is :2
The 1 no. is :1
```

PRACTICAL – 11

- Aim: Write a program that generate 6*6 two-dimensional matrix, filled with 0's and 1's , display the matrix, check every raw and column have an odd number's of 1's.
- Solution:

```
import java.util.Scanner;
class practical_11 {
public static int[][] create_fill_matrix() {
int[][] matrix = new int[6][6];
for (int i = 0; i < 6; i++) {
for (int j = 0; j < 6; j++) {
```

```
matrix[i][j] = (int) ((Math.random() * 5) % 2);
}
}
return matrix;
}
public static void displayMatrix(int[][] matrix) {
System.out.print("\nMatrix Values \n");
for (int i = 0; i < 6; i++) {
for (int j = 0; j < 6; j++) {
System.out.print(matrix[i][j] + " ");
}
System.out.println();
}
}
public static void main(String[] args) {
int my_matrix[][];
int i, j, cnt;
my_matrix = create_fill_matrix();
displayMatrix(my_matrix);
System.out.println("\nRows Having ODD no of 1s");
for (i = 0; i < 6; i++) {
cnt = 0;
for (j = 0; j < 6; j++) {
if (my_matrix[i][j] == 1) {
cnt++;
}
}
if (cnt % 2 != 0) {
System.out.println("Row - " + (i + 1) + " have ODD no of 1s");
}
}
System.out.println("\nColumns Having ODD no of 1s");
for (i = 0; i < 6; i++) {
cnt = 0;
for (j = 0; j < 6; j++) {
if (my_matrix[j][i] == 1) {
cnt++;
}
}
if (cnt % 2 != 0) {
System.out.println("Column - " + (i + 1) + " have ODD no of 1s");
}
}
}
}
```

- **Output:**

```
M:\java_practical>javac practical11.java

M:\java_practical>java practical11

Matrix
0 1 0 0 1 0
0 0 1 1 0 0
0 1 1 0 0 0
0 0 0 0 0 1
0 1 1 1 0 1
1 1 0 1 1 1

Rows Having ODD no of 1s :
   Row 4 have ODD no of 1s
   Row 6 have ODD no of 1s

Columns Having ODD no of 1s :
   Column 1 have ODD no of 1s
   Column 3 have ODD no of 1s
   Column 4 have ODD no of 1s
   Column 6 have ODD no of 1s
```

PRACTICAL – 12

- Aim: Write a program that creates a Random object with seed 1000 and displays the first 100 random integers between 1 and 49 using the NextInt (49) method.
- Solution:

```
import java.util.Random;
class practical_12 {
public static void main(String[] args) {
Random random = new Random(1000);
for (int i = 0; i < 100; i++) {
System.out.format("%5d", random.nextInt(49));
if ((i + 1) % 10 == 0) {
System.out.println();
}
}
}
}
```

- Output:

```
M:\java_practical>javac practical12.java

M:\java_practical>java practical12
  40   1  17   0  46  46   4  33   9  40  25  13  17  29  30  31  42  35  48  40
  11  13  10   0  38   9   0  10  35  10  14  26  34  35  31  43  47  35   2  33
  16  48  45  43   5  29   1  35   0  25  28  42  25   2  33  30  18  27   4  28
  31  35   9  13  33  12  18  36  39   7  17  31  21  26  47  39  11  40  11  26
  48  26  27  32  19  30  26   4   7  40   9  41   8  37   3  34  10  36   4  21
```

PRACTICAL – 13

- Aim: Write a program for calculator to accept an expression as a string in which the operands and operator are separated by zero or more spaces.
- Solution:

```
import java.util.Scanner;
class practical_13 {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
System.out.print("Enter Equation : ");
String string = sc.nextLine();
String a = string.replaceAll(" ", "");
if (a.length() < 3) {
System.out.println(
"Minimum 2 Opearator and 1 Opearand Required");
System.exit(0);
}
int result = 0;
int i = 0;
while (a.charAt(i) != '+' && a.charAt(i) != '-' && a.charAt(i) != '*' &&
a.charAt(i) != '/') {
i++;
}
switch (a.charAt(i)) {
case '+':
result = Integer.parseInt(a.substring(0, i)) +
Integer.parseInt(a.substring(i + 1, a.length()));
break;
case '-':
result = Integer.parseInt(a.substring(0, i)) -
Integer.parseInt(a.substring(i + 1, a.length()));
break;
case '*':
result = Integer.parseInt(a.substring(0, i)) *
Integer.parseInt(a.substring(i + 1, a.length()));
break;
case '/':
result = Integer.parseInt(a.substring(0, i)) /
Integer.parseInt(a.substring(i + 1, a.length()));
```

```
break;
}
System.out.println(a.substring(0, i) + ' ' + a.charAt(i) + ' ' +
a.substring(i + 1, a.length())
+ " = " + result);
}
}
```

- Output:

```
M:\java_practical>javac practical13.java

M:\java_practical>java practical13
Enter Equation : 3+4
3 + 4 = 7

M:\java_practical>java practical13
Enter Equation : 3   +   4
3 + 4 = 7
```

PRACTICAL – 14

- Aim: Write a program that creates an Array List and adds a Loan object , a Date object , a string, and a Circle object to the list, and use a loop to display all elements in the list by invoking the object's to String() method.
- Solution:

```
import java.util.ArrayList;
import java.util.Date;
class practical_14 {
public static void main(String[] args) {
ArrayList<Object> arr_list = new ArrayList<Object>();
arr_list.add(new Loan(10000));
arr_list.add(new Date());
arr_list.add(new String("GTU Practica-l4"));
arr_list.add(new Circle(3.45));
for (int i = 0; i < arr_list.size(); i++) {
System.out.println((arr_list.get(i)).toString());
}
}
}
class Circle {
double radius;
Circle(double r) {
this.radius = r;
}
public String toString() {
return "Circle with Radius " + this.radius;
}
```

```
}
class Loan {
double amount;
Loan(double amt) {
this.amount = amt;
}
public String toString() {
return "Loan with Amount " + this.amount;
}
}
```

- Output:

```
M:\java_practical>javac practical14.java

M:\java_practical>java practical14
Loan with Amount 56783.0
Wed Apr 27 22:00:15 IST 2022
 GEC GANDHINAGAR
Circle with Radius 5.56
```

PRACTICAL – 15

- Aim: Write the bin2Dec (string binary String) method to convert a binary string into a decimal number. Implement the bin2Dec method to throw a NumberFormatException if the string is not a binary string.
- Solution:

```
import java.util.Scanner;
class practical_15 {
public static int bin2Dec(String binaryString) throws NumberFormatException {
int decimal = 0;
int strLength = binaryString.length();
for (int i = 0; i < strLength; i++) {
if (binaryString.charAt(i) < '0' || binaryString.charAt(i) > '1') {
throw new NumberFormatException("The sc String is not Binary");
}
decimal += (binaryString.charAt(i) - '0') * Math.pow(2, strLength - 1 - i);
}
return decimal;
}
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
System.out.print("Enter Binary Value : ");
String str = sc.nextLine();
try {
System.out.println("Value = " + bin2Dec(str));
} catch (NumberFormatException e) {
System.out.println(e);
}
```

```
}
}
```

❖ Output:

```
M:\java_practical>javac practical15.java

M:\java_practical>java practical15
Enter Binary Value : 1001
Value = 9
```

PRACTICAL – 16

❖ Aim: Write a program that prompts the user to enter a decimal number and displays the number in a fraction.

❖ Solution:

```
import java.util.Scanner;
public class practical_16 {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
double number;
System.out.println("Enter the Decimal Number:- ");
number = sc.nextDouble();
int decimal_number = (int)number;
System.out.println("The Decimal number is " + decimal_number);
if (number-decimal_number < 0) {
System.out.println("The Fraction Number is : " + -(numberdecimal_
number) );
}
else
System.out.println("The Fraction Number is : " + (number-decimal_number) );
}
}import java.util.Scanner;
public class practical_16 {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
double number;
System.out.println("Enter the Decimal Number:- ");
number = sc.nextDouble();
int decimal_number = (int)number;
System.out.println("The Decimal number is " + decimal_number);
if (number-decimal_number < 0) {
System.out.println("The Fraction Number is : " + -(numberdecimal_
number) );
}
else
System.out.println("The Fraction Number is : " + (number-decimal_number) );
}
}
```

- Output:

```
Enter a Decimal Number: 0.25
The Fraction Number is 1/4
```

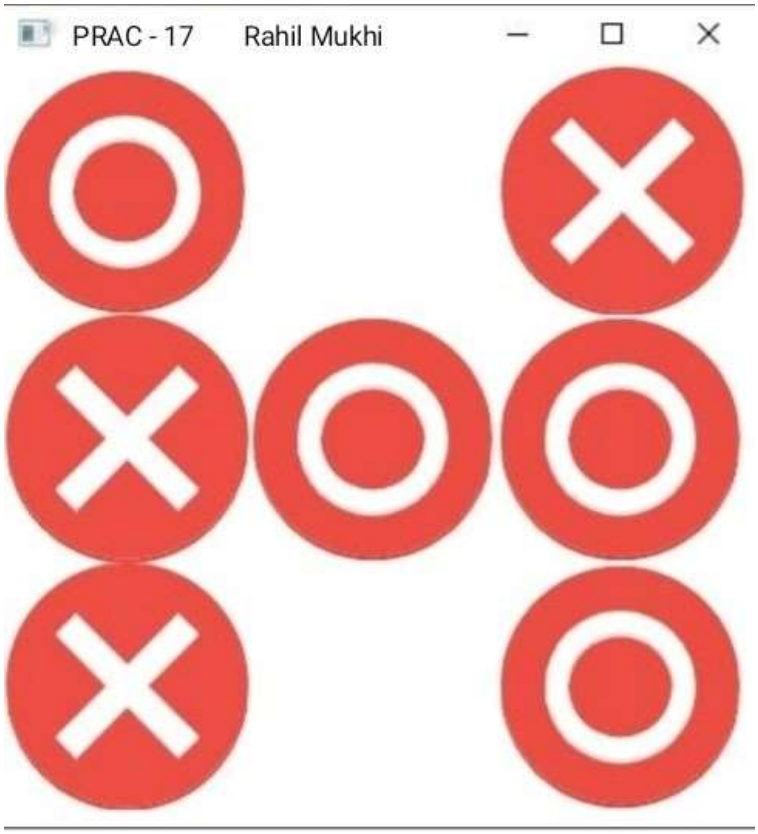
PRACTICAL – 17

- Aim: Write a program that displays a tic-tac-toe board. A cell may be X, O, or empty. What to display at each cell is randomly decided. The X and O are images in the files X.gif and O.gif
- Solution:

```
package sample;
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.stage.Stage;
import javafx.scene.image.Image;
import javafx.scene.image.ImageView;
import javafx.scene.layout.GridPane;
import javafx.scene.layout.VBox;
public class Main extends Application {
@Override
public void start(Stage primaryStage) throws Exception {
primaryStage.setTitle("Tic-Tac-Toe by Shlok Jadeja");
GridPane gridPane = new GridPane();
for (int i = 0; i < 3; i++) {
for (int j = 0; j < 3; j++) {
int n = (int) (Math.random() * 3);
if (n == 0)
gridPane.add(createX(), i, j);
else if (n == 1)
gridPane.add(createO(), i, j);
else
continue;
}
}
Scene primaryScene = new Scene(gridPane, 300, 300);
primaryStage.setScene(primaryScene);
primaryStage.show();
}
public VBox createX() {
Image imageX = new
Image("C:\\Users\\SHLOK\\IdeaProjects\\practical_17\\x.gif");
ImageView imageViewX = new ImageView(imageX);
VBox xBox = setProp(imageViewX);
return xBox;
}
public VBox createO() {
Image imageO = new
```

```
Image("C:\\Users\\SHLOK\\IdeaProjects\\practical_17\\O.gif");
ImageView imageViewO = new ImageView(imageO);
VBox oBox = setProp(imageViewO);
return oBox;
}
public VBox setProp(ImageView iv) {
iv.setFitHeight(50);
iv.setFitWidth(50);
iv.setPreserveRatio(true);
VBox vBox = new VBox();
vBox.getChildren().add(iv);
vBox.setStyle("-fx-border-color: orange");
return vBox;
}
public static void main(String[] args) {
launch(args);
}
}
```

- Output:



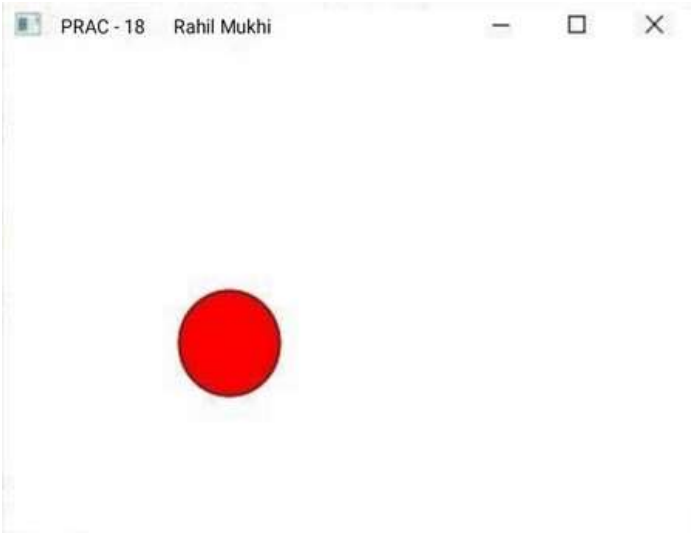
PRACTICAL – 18

- Aim: Write a program that moves a circle up, down, left or right using arrow keys.
  Solution:
- Solution:

```
package sample;
import javafx.application.Application;
import javafx.scene.Scene;
```

```
import javafx.scene.shape.Circle;
import javafx.scene.layout.Pane;
import javafx.geometry.Insets;
import javafx.stage.Stage;
public class Main extends Application
{
@Override
public void start(Stage primaryStage) {
Pane pane = new Pane();
pane.setPadding(new Insets(30, 30, 30, 30));
Circle circle = new Circle(30, 30, 30);
pane.getChildren().add(circle);
pane.setOnKeyPressed(e -> {
switch (e.getCode()) {
case UP : circle.setCenterY(circle.getCenterY() >
circle.getRadius() ? circle.getCenterY() - 15 :
circle.getCenterY()); break;
case DOWN : circle.setCenterY(circle.getCenterY() <
pane.getHeight() - circle.getRadius() ?
circle.getCenterY() + 15 : circle.getCenterY());
break;
case LEFT : circle.setCenterX(circle.getCenterX() >
circle.getRadius() ? circle.getCenterX() - 15 :
circle.getCenterX()); break;
case RIGHT : circle.setCenterX(circle.getCenterX() <
pane.getWidth() - circle.getRadius() ?
circle.getCenterX() + 15: circle.getCenterX());
}
});
Scene scene = new Scene(pane, 200, 200);
primaryStage.setTitle("Pracctical -18 by shlok Jadeja");
primaryStage.setScene(scene);
primaryStage.show();
pane.requestFocus();
}
public static void main(String[] args) {
launch(args);
}
}
```

- Output:

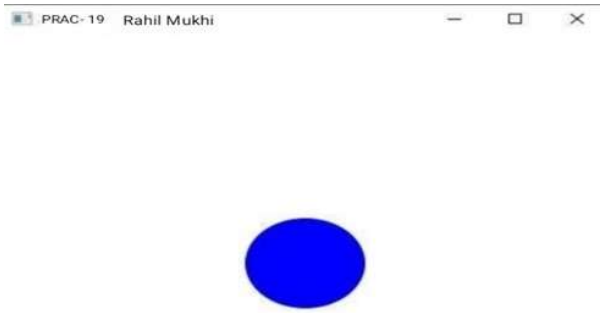


PRACTICAL – 19

- Aim: Write a program that displays the color of a circle as red when the mouse button is pressed and as blue when the mouse button is released.
- Solution:

```
package sample;
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.layout.StackPane;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;
import javafx.stage.Stage;
public class Main extends Application
{
@Override
public void start(Stage primaryStage) {
double width = 400;
double height = 400;
Circle c = new Circle(width / 2, height / 2, Math.min(width, height) / 10,
Color.WHITE);
c.setStroke(Color.BLACK);
StackPane pane = new StackPane(c);
primaryStage.setScene(new Scene(pane, width, height));
pane.setOnMousePressed(e -> c.setFill(Color.RED));
pane.setOnMouseReleased(e -> c.setFill(Color.BLUE));
primaryStage.setTitle("Click circle..");
primaryStage.show();
}
public static void main(String[] args) {
launch(args);
}
}
```

- **Output:**



PRACTICAL – 20

- Aim: Write a GUI program that use button to move the message to the left and right and use the radio button to change the color for the message displayed.
- Solution:

```
package sample;
import javafx.application.Application;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.geometry.Pos;
import javafx.scene.control.Button;
import javafx.scene.layout.HBox;
import javafx.scene.layout.Pane;
import javafx.scene.layout.BorderPane;
import javafx.scene.text.Text;
import javafx.scene.control.RadioButton;
import javafx.scene.control.ToggleGroup;
import javafx.scene.paint.Color;
public class Main extends Application
{
protected Text text = new Text(50, 50, "Hello Everyone This is Shlok Jadeja");
@Override // Override the stage method in the Application class
public void start(Stage primaryStage) {
HBox paneForButtons = new HBox(20);
Button btLeft = new Button("<=");
Button btRight = new Button("=>");
paneForButtons.getChildren().addAll(btLeft, btRight);
paneForButtons.setAlignment(Pos.CENTER);
BorderPane pane = new BorderPane();
pane.setBottom(paneForButtons);
HBox paneForRadioButtons = new HBox(20);
RadioButton rbRed = new RadioButton("Red");
RadioButton rbYellow = new RadioButton("Yellow");
RadioButton rbBlack = new RadioButton("Black");
RadioButton rbOrange = new RadioButton("Orange");
```

```
RadioButton rbGreen = new RadioButton("Green");
paneForRadioButtons.getChildren().addAll(rbRed, rbYellow,
rbBlack, rbOrange, rbGreen);
ToggleGroup group = new ToggleGroup();
rbRed.setToggleGroup(group);
rbYellow.setToggleGroup(group);
rbBlack.setToggleGroup(group);
rbOrange.setToggleGroup(group);
rbGreen.setToggleGroup(group);
Pane paneForText = new Pane();
paneForText.setStyle("-fx-border-color: black");
paneForText.getChildren().add(text);
pane.setCenter(paneForText);
pane.setTop(paneForRadioButtons);
btLeft.setOnAction(e -> text.setX(text.getX() - 10));
btRight.setOnAction(e -> text.setX(text.getX() + 10));
rbRed.setOnAction(e -> {
if (rbRed.isSelected()) {
text.setFill(Color.RED);
}
});
rbYellow.setOnAction(e -> {
if (rbYellow.isSelected()) {
text.setFill(Color.YELLOW);
}
});
rbBlack.setOnAction(e -> {
if (rbBlack.isSelected()) {
text.setFill(Color.BLACK);
}
});
rbOrange.setOnAction(e -> {
if (rbOrange.isSelected()) {
text.setFill(Color.ORANGE);
}
});
rbGreen.setOnAction(e -> {
if (rbGreen.isSelected()) {
text.setFill(Color.GREEN);
}
});
// Create a scene and place it in the stage
Scene scene = new Scene(pane, 450, 150);
primaryStage.setTitle("Practical-20 by Shlok Jadeja"); // Set the stage
title
primaryStage.setScene(scene); // Place the scene in the stage
primaryStage.show(); // Display the stage
}
```

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

- Output:

PRACTICAL – 21

- Aim: Write a program to create a file name 123.txt, if it does not exist. Append a new data to it if it already exist. write 150 integers created randomly into the file using Text I/O. Integers are separated by space.
- Solution:

```
import java.io.*;
import java.util.Scanner;
public class practical_21 {
public static void main(String[] args) {
try (
PrintWriter pw = new PrintWriter(new FileOutputStream(new File("123.txt"),
true));
) {
for (int i = 0; i < 150; i++) {
pw.print((int)(Math.random() * 100) + " ");
}
}
catch (FileNotFoundException fnfe) {
System.out.println("Cannot create the file.");
fnfe.printStackTrace();
}
}
}
```

- Output:

```
M:\java_practical>javac practical21.java

M:\java_practical>java practical21

M:\java_practical>type 123.txt
143 81 45 87 149 60 79 48 81 75 38 144 38 121 25 125 70 62 64 26 105 126 79 23 91 37 142 102 106 111 30 95 22 37 88 37 1
32 57 43 142 25 14 1 53 54 83 83 125 33 1 5 136 125 75 44 108 75 23 85 6 44 96 129 65 60 15 101 134 34 126 49 93 106 148
 126 97 126 107 102 111 97 132 87 33 58 105 58 94 78 57 118 50 116 147 67 13 103 51 77 139 0 68 26 139 12 86 18 22 15 57
 36 71 96 79 144 45 93 46 127 67 21 82 3 81 134 74 1 100 69 33 30 35 74 139 45 15 40 108 24 22 113 81 93 21 78 56 133 10
 32 64
```

PRACTICAL – 22

- **Aim: Write a recursive method that returns the smallest integer in an array. Write a test program that prompts the user to enter an integer and display its product.**
- **Solution:**

```
import java.util.Scanner;
import java.util.*;
class practical_22 {
public static int MinimumUSingRec(int Array[], int num)
{
// if size = 0 means whole array
// has been traversed
if(num == 1)
return Array[0];
return Math.min(Array[num-1], MinimumUSingRec(Array, num-1));
}
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
int product = 1;
System.out.println("Enter five integers: ");
int[] list = new int[5];
for (int i = 0; i < list.length; i++) {
list[i] = sc.nextInt();
product *= list[i];
}
System.out.println("The Product of elements is " + product);
System.out.println("The minimum number of these Five number is " +
MinimumUSingRec(list, 5));
}
}
```

- **Output:**

```
M:\java_practical>java practical22.java
array elements :
 34 45 11 22 33
 minimum value from array is  : 11
Enter Integer : 45
The multiplication of 11 and 45 is = 495
```

PRACTICAL – 23

- **Aim: Write a generic method that returns the minimum elements in a two dimensional array.**
- **Solution:**

```
import java.util.*;
public class practical_23 {
public static void main(String[] args) {
Integer list[][] = new Integer[3][3];
Scanner sc = new Scanner(System.in);
int value = 1;
for (int i = 0; i < list.length; i++) {
for (int j = 0; j < list[i].length; j++) {
System.out.println("Enter " + value + " Number:-");
list[i][j] = sc.nextInt();
value++;
}
}
System.out.println("Minimum no is = " + min(list));
}
public static <E extends Comparable<E>> E min(E list[][]) {
E min = list[0][0];
for (E[] elements : list) {
for (E element : elements) {
if (element.compareTo(min) < 0) {
min = element;}}}
return min;}}
```

- **Output:**

```
M:\java_practical>javac practical23.java

M:\java_practical>java practical23
Enter element
10 11 12
13 14 15
17 18 19
Minimum value is = 10
```

PRACTICAL – 24

- **Aim: Define MYPriorityQueue class that extends Priority Queue to implement the Cloneable interface and implement the clone() method to clone a priority queue.**
- **Solution:**

```
import java.util.PriorityQueue;
public class practical_24 {
public static void main(String[] args) {
MyPriorityQueue<String> queue = new MyPriorityQueue<>();
queue.offer("1");
queue.offer("2");
queue.offer("3");
```

```
MyPriorityQueue<String> queue1 = null;
try {
queue1 = (MyPriorityQueue<String>)(queue.clone());
} catch (CloneNotSupportedException e) {
e.printStackTrace();
}
System.out.print(queue1);
}
static class MyPriorityQueue<E> extends PriorityQueue<E> implements Cloneable {
@Override
public Object clone() throws CloneNotSupportedException {
MyPriorityQueue<E> clone = new MyPriorityQueue<>();
this.forEach(clone::offer);
return clone;
}
}
}
```

- **Output:**

```
M:\java_practical>javac practical24.java

M:\java_practical>java practical24
Queue1: 10 20 30
Queue2: 10 20 30
```

PRACTICAL – 25

- **Aim: Write a program that reads words from a text file and displays all the nonduplicate words in descending order. The text file is passed as a command-line argument.**
- **Solution:**

```
import java.io.*;
import java.util.*;
class practical_25
{
public static void main(String[] args) throws Exception
{
File f=new File("Demo.txt");
BufferedReader br=new BufferedReader(new FileReader(f));
StringBuffer buffer=new StringBuffer();
String str;
while((str=br.readLine())!=null){
buffer.append(str);
buffer.append(" ");
}
ArrayList list=new ArrayList();
StringTokenizer st = new StringTokenizer(buffer.toString().toLowerCase());
while(st.hasMoreTokens()) {
String s = st.nextToken();
```

```
list.add(s);
}
HashSet set = new HashSet(list);
List arrayList = new ArrayList(set);
Collections.sort(arrayList);
for (Object ob : arrayList)
System.out.println(ob.toString());
}
}
```

- **Output:**