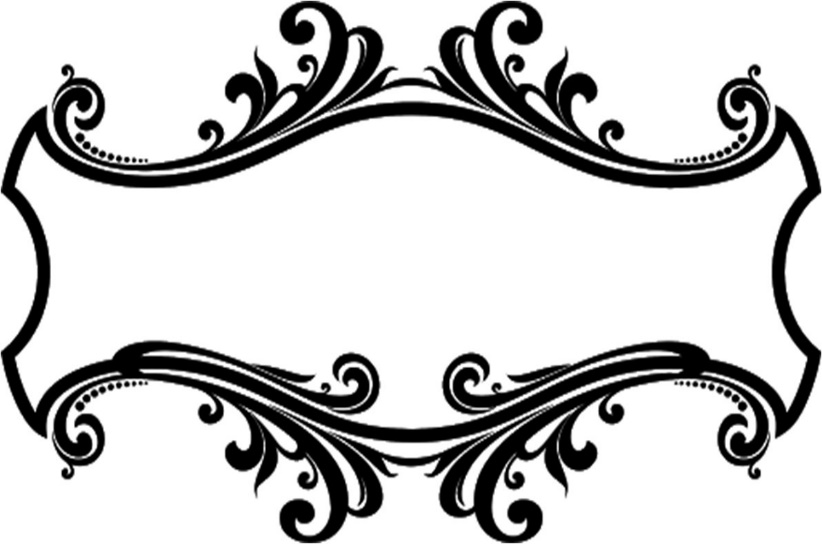
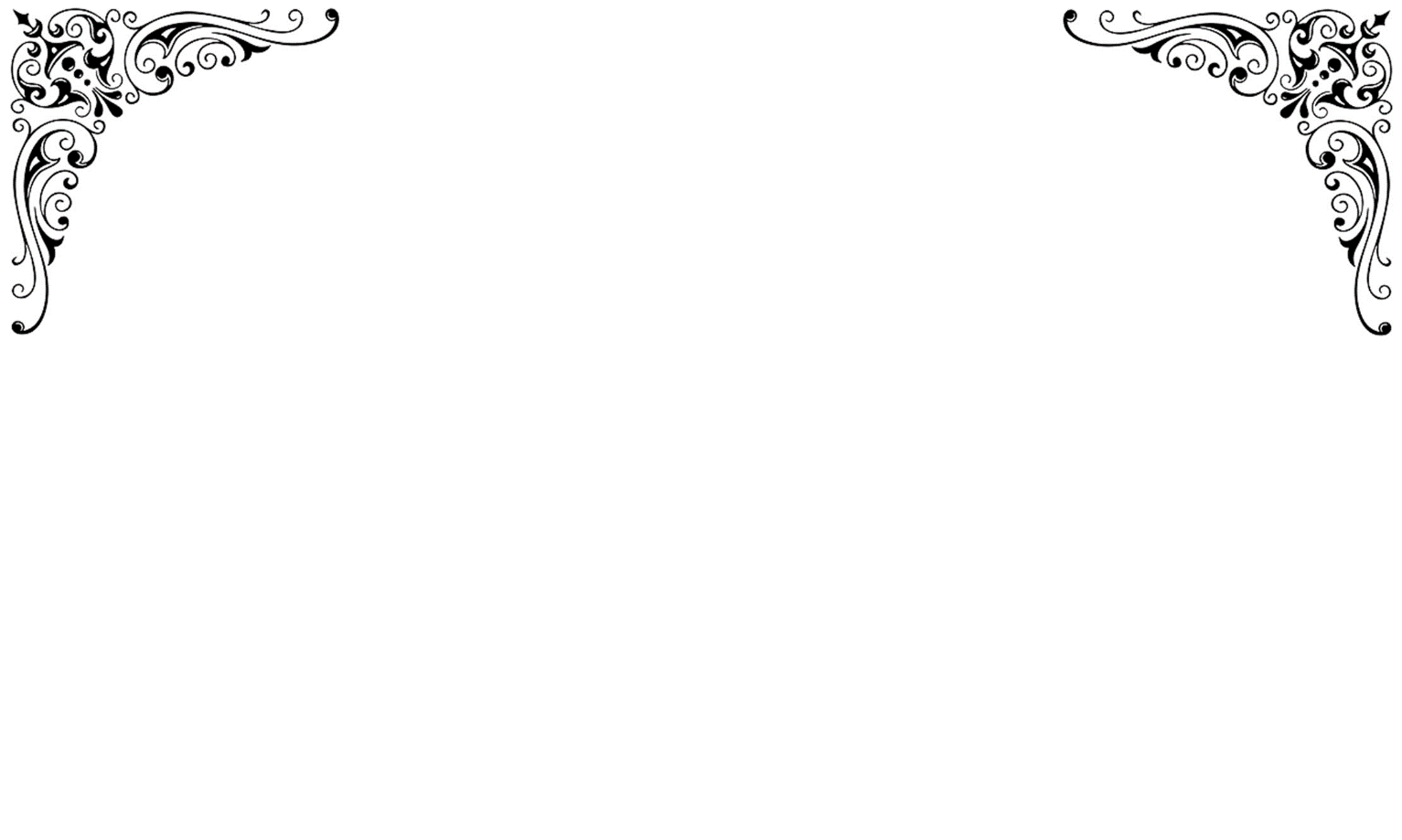
.



SESSION 2023-24

Java

(Java Programming Language)

Lab File

**COURSE:- BCA**

**ROLL NO :- 41221139**

**SUBMITTED BY :- SUBMITTED TO:-**

**Sachin Rajbhar Mrs. Shilpa Ma’am**

|  |  |  |  |
| --- | --- | --- | --- |
| I N D E X | | | |
| S.NO | **P R A C T I C A L S** | **DATE** | **SIGN.** |
| 1 | WAP to print hello world |  |  |
| 2 | WAP to find the largest of two numbers |  |  |
| 3 | WAP to find number entered by user is even or odd |  |  |
| 4 | WAP to find year entered by user is leap or not |  |  |
| 5 | WAP to find factorial of the number entered by user |  |  |
| 6 | WAP to print n numbers of Fibonacci series, n is given by user |  |  |
| 7 | WAP to print the sum, average and product of three numbers |  |  |
| 8 | WAP to calculate sum and product of all digit of an integer entered by user (5 digit number ) using class and object |  |  |
| 9 | WAP to print the area and parameter of 2 rectangle having sides 15,20 and 50,125 respectively |  |  |
| 10 | WAP to create an account details for a user with two methods Deposit and Withdraw { hint: add two more methods insert() & display() taking the details of the user as   * Acc no. * Name * Amount }   using constructor |  |  |
| 11 | Create a class named 'Company\_database' having the following members: Data members 1 - Name 2 - Age 3 - Phone number 4 – Company\_ID 5 - Salary It also has a method named 'printSalary' which prints the salary of the members.Two classes 'Employee' and 'Manager' inherits the 'Company Database' class. The 'Employee' and 'Manager' classes have data members 'core\_area' and 'Branch' respectively. Now, assign name, age, phone number, Company\_ID and salary to an employee and a manager by making an object of both of these classes and print the same. |  |  |
| 12 | Create a class named 'Shape' with a method to print "This is This is shape". Then create two other classes named 'Rectangle', 'Circle' inheriting the Shape class, both having a method to print "This is rectangular shape" and "This is circular shape" respectively. Create a subclass 'Square' of 'Rectangle' having a method to print "Square is a rectangle". Now call the method of 'Shape' and 'Rectangle' class by the object of 'Square' class. |  |  |
| 13 | Develop a Java Program to create an abstract class named Shape that contains two integers and an empty method named printArea(). 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. |  |  |
| 14 | A superclass named “Shapes” has a method called “area()”. Subclasses of “Shapes” can be “Triangle”, “circle”, “Rectangle”, etc. Each subclass has its way of calculating area. Using Inheritance and Polymorphism means, the subclasses can use the “area()” method to find the area’s formula for that shape. |  |  |
| 15 | WAP to find calculate the simple interest for three different banks giving ROI as 8%, 9% and 7.2% correspondingly making RBI as the Parent Class. Implementing the concept of method Overriding. Taking Principal amount and number of years from user as input |  |  |
| 16 | Wap to show implementation of Multi threading |  |  |
| 17 | Wap to show synchronization in multi threading by printing table of two and Five |  |  |
| 18 | Wap to Draw a string “welcome to Applet. It is my first Applet program”in Applet |  |  |
| 19 | WAP in applet to draw four different shapes and set color, Fill color in all. |  |  |
| 20 | WAP in Applet to set background and foreground color. |  |  |
| 21 | WAP in Applet to Add an image |  |  |
| 22 | WAP to draw a smiley using APPLET |  |  |
| 23 | WAP to make a registration form using AWT |  |  |
| 24 | WAP to Java Program to Create 2 Radio Buttons and Display Selected Button Label. |  |  |
| 25 | WAP to Create a calculator using Swings. |  |  |
| 26 | WAP to get Hostname IP address and username |  |  |
| 27 | WAP to implement client side and server-side coding for creating  connection and maintain communication between client and  server |  |  |
| 28 | WAP to implement JDBC concept by creating a database in  XAMP and connect it using java. |  |  |
| 29 | Project Wok : Create a puzzle game using Swings through AWT api with event handling. |  |  |
|  |  |  |  |

**PRACTICAL 1**

Q: WAP to print hello world

**CODE:**

public class main {

    public static void main(String[] args) {

        System.out.println("Hello Java");

   }

}

**OUTPUT:**

****

**PRACTICAL 2**

Q: WAP to find the largest of two numbers

**CODE:**

import java.util.Scanner;

public class largest2 {

   public static void main(String[] args)

   {

      int num1, num2, largest;

      Scanner scan = new Scanner(System.in);

      System.out.print("Enter the First Number: ");

      num1 = scan.nextInt();

      System.out.print("Enter the Second Number: ");

      num2 = scan.nextInt();

      scan.close();

    if(num1>num2){

         largest = num1;

         System.out.println("\nLargest = " +largest);

    }else if(num1<num2){

        largest = num2;

        System.out.println("\nLargest = " +largest);

    }

    else

    {

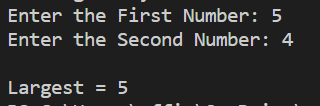
        System.out.println("\nBoth are same");

    }

   }

}

**OUTPUT:**

****

**PRACTICAL 3**

Q: WAP to find no. entered by user is even or odd

**CODE:**

import java.util.Scanner;

public class evenodd {

    public static void main(String[] args) {

        Scanner reader = new Scanner(System.in);

        System.out.println("Enter a number: ");

        int num = reader.nextInt();

        reader.close();

        if(num % 2 == 0)

            System.out.println(num + " is even");

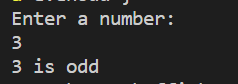
        else

            System.out.println(num + " is odd");

    }

}

**OUTPUT:**

****

**PRACTICAL 4**

Q: WAP to find year entered by user is leap or not

**CODE:**

import java.util.Scanner;

public class LeapYear {

   public static void main(String[] args){

      int year;

      System.out.print("Enter an Year : ");

      Scanner sc = new Scanner(System.in);

      year = sc.nextInt();

      sc.close();

      if (((year % 4 == 0) && (year % 100!= 0)) || (year%400 == 0))

         System.out.println( year+" is a leap year");

      else

         System.out.println( year+" is not a leap year");

   }

}

**OUTPUT:**

****

**PRACTICAL 5**

Q: WAP to find factorial of the number entered by user

**CODE:**

import java.util.\*;

public class Factorial{

     public static void main(String []args)

     {

        Scanner sc=new Scanner(System.in);

        System.out.print("Enter the number: ");

        int num=sc.nextInt();

        sc.close();

        int fact=1;

        for(int i=1;i<=num;i++)

        {

            fact=fact\*i;

        }

        System.out.println("Factorial of the number: "+fact);

     }

}

**OUTPUT:**

****

**PRACTICAL 6**

Q: WAP to print n numbers of Fibonacci series, n is given by user

**CODE:**

import java.util.Scanner;

class Fibonacci {

    // Function to print N Fibonacci Number

    static void Fibonacciseries(int N)

    {

        int num1 = 0, num2 = 1;

        int counter = 0;

        // Iterate till counter is N

        while (counter < N) {

            // Print the number

            System.out.print(num1 + " ");

            // Swap

            int num3 = num2 + num1;

            num1 = num2;

            num2 = num3;

            counter = counter + 1;

        }

    }

    // Driver Code

    public static void main(String args[])

    {

        // Given Number N

        System.out.println(“Enter value of N:“);

        Scanner sc = new Scanner(System.in);

        int N = sc.nextInt();

        sc.close();

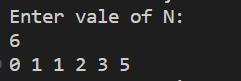
        // Function Call

        Fibonacciseries(N);

    }

}

**OUTPUT:**

****

**PRACTICAL 7**

Q: Print the sum, average and product of three numbers

**CODE:**

class  sum {

    int a , b, c ;

    void insert(int x, int y,int z) {

        a= x ;

        b=y ;

         c= z ;

    }

    void calculatesum (){

        System.out.println("Sum is :"+a+b+c);

    }

    void calculateproduct(){

        System.out.println("Product is :"+a\*b\*c);

    }

    void calculateavg(){

        System.out.println(("Average is :"+(a+b+c)/3));

    }

}

public class sumAvgProOf3 {

    public static void main(String[] args){

        sum s1 =new sum();

        sum s2 =new sum();

        sum s3 =new sum();

        s1.insert(10 ,34,56);

        s2.insert(23,45,67);

        s3.insert(10,20,30);

        s1.calculatesum();

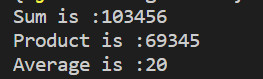
        s2.calculateproduct();

        s3.calculateavg();

    }

}

**OUTPUT:**

****

**PRACTICAL 8**

Q: WAP to calculate sum and product of all digit of an integer entered by user (5 digit number ) using class and object

**CODE:**

import  java.util.Scanner ;

class Calculate{

    int num;

    void result(int n){

        int digit;

        int sum =0;

        int product=1;

        for(num=n; num>0;){

            digit = num%10;

            sum = sum +digit;

            product = product \* digit;

            num = num/10;

        }

        System.out.println("sum 5 digit  :"+ sum);

        System.out.println("product 5 digits :"+ product);

    }

}

public class digitsSumPro {

    public static void main(String [] args) {

        Calculate c1=new Calculate();

        Scanner scan =new Scanner(System.in);

        System.out.println("Enter a number :");

        int n =scan.nextInt();

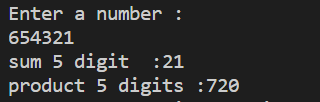
        c1.result(n);

        scan.close();

    }

}

**OUTPUT:**

****

**PRACTICAL 9**

Q: WAP to print the area and perimeter of 2 rectangle having sides 15,20 and 50,125 respectively

**CODE:**

class Rectangle {

    int length;

    int breadth;

    void insert(int l, int b) {

        length = l;

        breadth = b;

    }

    int calcArea() {

        return length \* breadth;

    }

    int calcPeri() {

        return 2 \*(length + breadth);

    }

}

public class rectAreaPeri {

    public static void main(String[] args) {

        Rectangle r1 = new Rectangle();

        Rectangle r2 = new Rectangle();

        r1.insert(15, 20);

        r2.insert(50,125);

        System.out.println("Area of rectangle1 is: " + r1.calcArea());

        System.out.println("Perimeter of rectangle1 is: " + r1.calcPeri());

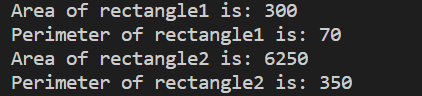
        System.out.println("Area of rectangle2 is: " + r2.calcArea());

        System.out.println("Perimeter of rectangle2 is: " + r2.calcPeri());

    }

}

**OUTPUT:**

****

**PRACTICAL 10**

Q: WAP to create an account details for a user with two methods Deposit and Withdraw { hint: add two more methods insert() & display() taking the details of the user as

* Acc no.
* Name
* Amount

using constructor

**CODE:**

import java.util.Scanner;

class Account {

    int accno;

    String name;

    int amount=0;

    Account (int a, String n, int amt) {

        accno = a;

        name = n;

        amount = amt;

    }

    void insert(int a, String n, int amt) {

        accno = a;

        name = n;

        amount = amt;

    }

    void deposit(int amt) {

        amount = amount + amt;

        System.out.println(amt + " deposited");

    }

    void withdraw(int amt) {

        if(amount < amt){

            System.out.println("Insufficient Balance");

        } else {

            amount = amount - amt;

            System.out.println(amt + " withdrawn");

        }

    }

    void display() {

        System.out.println("Name: " + name);

        System.out.println("Account number: " + accno);

        System.out.println("Ac Balance: " + amount);

    }

}

class userAcc {

    public static void main(String[] args) {

        Account a1 = new Account(1677, "Mr. Ravindra", 1000000000);

        Scanner sc = new Scanner(System.in);

        char ch;

        do {

            System.out.println("\n\nWelcome to the RAKA Bank\n");

            System.out.println("Please select an option:");

            System.out.println("1. Create a new account");

            System.out.println("2. Deposit money");

            System.out.println("3. Withdraw money");

            System.out.println("4. Check balance");

            System.out.println("5. Exit");

            System.out.println("\nEnter your choice: ");

            int choice = sc.nextInt();

            switch(choice) {

                case 1:

                    System.out.println("Enter your name: ");

                    String name = sc.next();

                    System.out.println("Enter your account number: ");

                    int accno = sc.nextInt();

                    System.out.println("Enter the amount you want to deposit: ");

                    int amt = sc.nextInt();

                    a1.insert(accno, name, amt);

                    System.out.println("Account created successfully");

                    break;

                case 2:

                    System.out.println("Enter the amount you want to deposit: ");

                    int deposit = sc.nextInt();

                    a1.deposit(deposit);

                    break;

                case 3:

                    System.out.println("Enter the amount you want to withdraw: ");

                    int withdraw = sc.nextInt();

                    a1.withdraw(withdraw);

                    break;

                case 4:

                    a1.display();

                    break;

                case 5:

                    System.out.println("Thank you for using RAKA Bank");

                    ch = 'n';

                    break;

                default:

                    System.out.println("Invalid choice");

            }

            System.out.println("Do you want to continue? (y/n)");

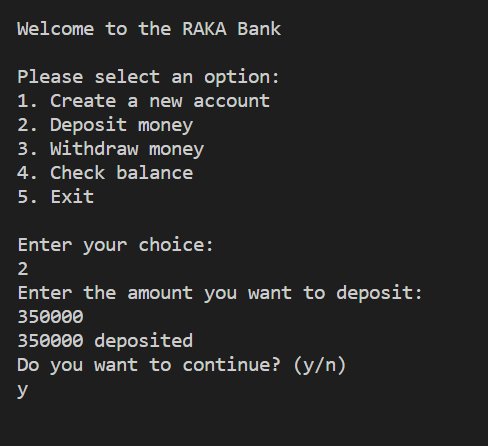
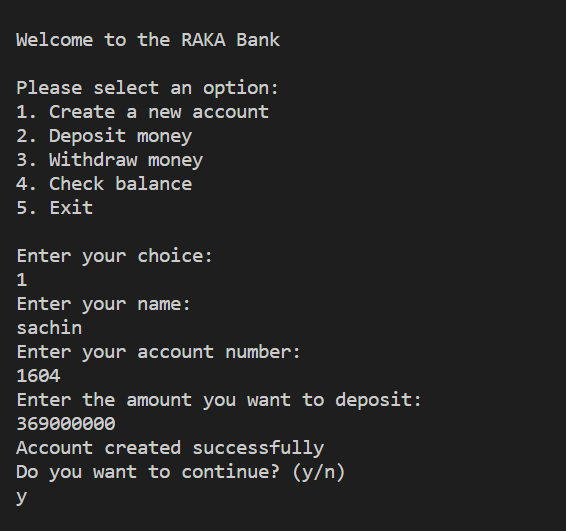
            ch = sc.next().charAt(0);

        } while (ch == 'y'|| ch == 'Y');

    }

}

**OUTPUT:**

****

**PRACTICAL 11**

Q: WAP to create a class named 'Company\_database'

**CODE:**

class company\_database {

    char name;

    int age, phno, company\_id, salary;

    void printSalary() {

        System.out.println("Salary is "+salary);

    }

}

class employee extends company\_database {

    char core\_area;

}

class manager extends company\_database {

    char branch;

}

public class company {

    public static void main(String[] args) {

        //object e1

        employee e1 = new employee();

        e1.name = 'A';

        e1.age = 20;

        e1.phno = 1234567890;

        e1.company\_id = 1;

        e1.salary = 10000000;

        e1.core\_area = 'B';

        e1.printSalary();

        //object m1

        manager m1 = new manager();

        m1.name = 'C';

        m1.age = 30;

        m1.phno = 1234567890;

        m1.company\_id = 2;

        m1.salary = 20000000;

        m1.branch = 'D';

        m1.printSalary();

    }

}

**OUTPUT:**

****

**PRACTICAL 12**

Q: WAP to create a class named 'Shape' with a method to print "This is shape"

**CODE:**

class shape {

    void prints() {

        System.out.println("this is a shape");

    }

}

class rectangle extends shape{

    void printr() {

        System.out.println("this is a rectangular shape");

    }

}

class circle extends shape {

    void print() {

        System.out.println(" this is a circular shape");

    }

}

class square extends rectangle {

    void print() {

        System.out.println("square is a rectangle");

    }

}

public class multiple {

    public static void main(String[] args) {

        square s1 = new square();

        s1.prints();

        s1.printr();

    }

}

**OUTPUT:**

****

**PRACTICAL 13**

Q: WAP in java to create a abstract class

**CODE:**

abstract class Shape {

    protected int length;

    protected int width;

    public Shape(int length, int width) {

        this.length = length;

        this.width = width;

    }

    abstract void printArea();

}

class Rectangle extends Shape {

    public Rectangle(int length, int width) {

        super(length, width);

    }

    @Override

    void printArea() {

        int area = length \* width;

        System.out.println("Area of Rectangle: " + area);

    }

}

class Triangle extends Shape {

    public Triangle(int length, int width) {

        super(length, width);

    }

    @Override

    void printArea() {

        int area = (length \* width) / 2;

        System.out.println("Area of Triangle: " + area);

    }

}

class Circle extends Shape {

    private double radius;

    public Circle(double radius) {

        super(0, 0); // length and width are not applicable for Circle

        this.radius = radius;

    }

    @Override

    void printArea() {

        double area = Math.PI \* radius \* radius;

        System.out.println("Area of Circle: " + area);

    }

}

public class MainAbstract {

    public static void main(String[] args) {

        Rectangle rectangle = new Rectangle(4, 5);

        rectangle.printArea();

        Triangle triangle = new Triangle(4, 5);

        triangle.printArea();

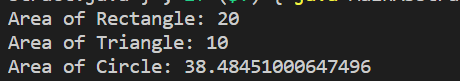
        Circle circle = new Circle(3.5);

        circle.printArea();

    }

}

**OUTPUT:**

****

**PRACTICAL 14**

Q: WAP to implement superclass & subclasses

**CODE:**

class Shape {

    public double area() {

        return 0; // Default implementation for unknown shapes

    }

}

class Rectangle extends Shape {

    private double length;

    private double width;

    public Rectangle(double length, double width) {

        this.length = length;

        this.width = width;

    }

    @Override

    public double area() {

        return length \* width;

    }

}

class Triangle extends Shape {

    private double base;

    private double height;

    public Triangle(double base, double height) {

        this.base = base;

        this.height = height;

    }

    @Override

    public double area() {

        return (base \* height) / 2;

    }

}

class Circle extends Shape {

    private double radius;

    public Circle(double radius) {

        this.radius = radius;

    }

    @Override

    public double area() {

        return Math.PI \* radius \* radius;

    }

}

public class MainSupSub {

    public static void main(String[] args) {

        Shape rectangle = new Rectangle(4, 5);

        System.out.println("Area of Rectangle: " + rectangle.area());

        Shape triangle = new Triangle(4, 5);

        System.out.println("Area of Triangle: " + triangle.area());

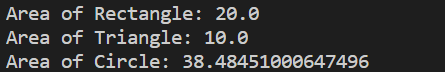
        Shape circle = new Circle(3.5);

        System.out.println("Area of Circle: " + circle.area());

    }

}

**OUTPUT:**

****

**PRACTICAL 15**

Q: WAP to find simple interest

**CODE:**

import java.util.Scanner;

class RBI {

    protected double rateOfInterest;

    public RBI(double rateOfInterest) {

        this.rateOfInterest = rateOfInterest;

    }

    public double calculateInterest(double principal, int years) {

        return (principal \* rateOfInterest \* years) / 100;

    }

}

class BankA extends RBI {

    public BankA(double rateOfInterest) {

        super(rateOfInterest);

    }

    @Override

    public double calculateInterest(double principal, int years) {

        return super.calculateInterest(principal, years);

    }

}

class BankB extends RBI {

    public BankB(double rateOfInterest) {

        super(rateOfInterest);

    }

    @Override

    public double calculateInterest(double principal, int years) {

        return super.calculateInterest(principal, years);

    }

}

class BankC extends RBI {

    public BankC(double rateOfInterest) {

        super(rateOfInterest);

    }

    @Override

    public double calculateInterest(double principal, int years) {

        return super.calculateInterest(principal, years);

    }

}

public class MainInterest {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the principal amount: ");

        double principal = scanner.nextDouble();

        System.out.print("Enter the number of years: ");

        int years = scanner.nextInt();

        RBI bankA = new BankA(8.0);

        double interestA = bankA.calculateInterest(principal, years);

        System.out.println("Simple interest for Bank A: " + interestA);

        RBI bankB = new BankB(9.0);

        double interestB = bankB.calculateInterest(principal, years);

        System.out.println("Simple interest for Bank B: " + interestB);

        RBI bankC = new BankC(7.2);

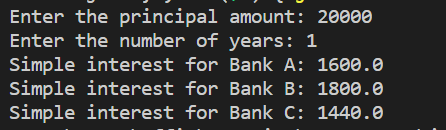
        double interestC = bankC.calculateInterest(principal, years);

        System.out.println("Simple interest for Bank C: " + interestC);

    }

}

**OUTPUT:**

****

**PRACTICAL 16**

Q: **Wap to show implementation of Multi threading**

**CODE:**

class MyThread extends Thread {

    public void run() {

        System.out.println("Thread: " + Thread.currentThread().getId() + " is running.");

    }

}

class MyRunnable implements Runnable {

    public void run() {

        System.out.println("Thread: " + Thread.currentThread().getId() + " is running.");

    }

}

public class MainThread {

    public static void main(String[] args) {

        // Creating and starting a thread using the Thread class

        MyThread thread1 = new MyThread();

        thread1.start();

        // Creating and starting a thread using the Runnable interface

        Thread thread2 = new Thread(new MyRunnable());

        thread2.start();

        // Creating and starting a thread using an anonymous inner class

        Thread thread3 = new Thread(new Runnable() {

            public void run() {

                System.out.println("Thread: " + Thread.currentThread().getId() + " is running.");

            }

        });

        thread3.start();

        // Creating and starting a thread using lambda expression

        Thread thread4 = new Thread(() -> {

            System.out.println("Thread: " + Thread.currentThread().getId() + " is running.");

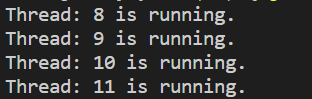
        });

        thread4.start();

    }

}

**OUTPUT:**

****

**PRACTICAL 17**

Q: **Wap to show synchronization in multi threading by printing table of two and Five**

**CODE:**

class TablePrinter {

    synchronized void printTable(int number) {

        for (int i = 1; i <= 10; i++) {

            System.out.println(number + " \* " + i + " = " + (number \* i));

            try {

                Thread.sleep(500); // Introducing a small delay for better visualization

            } catch (InterruptedException e) {

                e.printStackTrace();

            }

        }

    }

}

class TableThread extends Thread {

    private TablePrinter tablePrinter;

    private int number;

    public TableThread(TablePrinter tablePrinter, int number) {

        this.tablePrinter = tablePrinter;

        this.number = number;

    }

    public void run() {

        tablePrinter.printTable(number);

    }

}

public class MainSync {

    public static void main(String[] args) {

        TablePrinter tablePrinter = new TablePrinter();

        TableThread thread1 = new TableThread(tablePrinter, 2);

        TableThread thread2 = new TableThread(tablePrinter, 5);

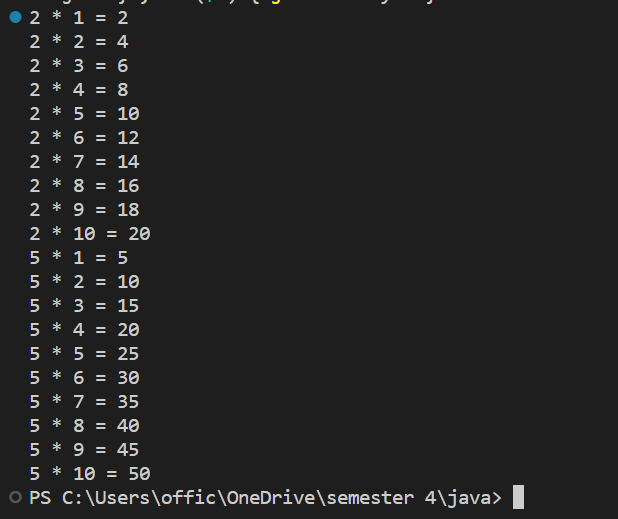
        thread1.start();

        thread2.start();

    }

}

**OUTPUT:**

****

**PRACTICAL 18**

Q: **Wap to Draw a string “welcome to Applet. It is my first Applet program”in Applet**

**CODE:**

//First

import java.applet.Applet;

import java.awt.Graphics;

public class First extends Applet{

public void paint(Graphics g){

g.drawString("welcome to applet",150,150);

}

}

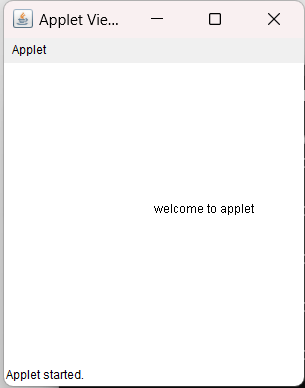
/\*

<applet code="First.class" width="300" height="300">

</applet>

\*/

**OUTPUT:**

****

**PRACTICAL 19**

Q: **WAP in applet to draw four different shapes and set color, Fill color in all.**

**CODE:**

import java. applet.Applet;

import java.awt.\*;

public class Second extends Applet{

public void paint (Graphics g) {

g.setColor (Color.red);

g.drawString("Welcome", 50, 50);

g.drawLine (20,30,20,300);

g.drawRect (200,250,100,100);

g.fillRect(170,100,30,30);

g.drawOval (70,200,30,30);

g.setColor(Color.pink);

g.fillOval (170,200,30,30);

g.drawArc (90,150,30,30,30,270);

g.fillArc(270,150,30,30,0,180);

}

}

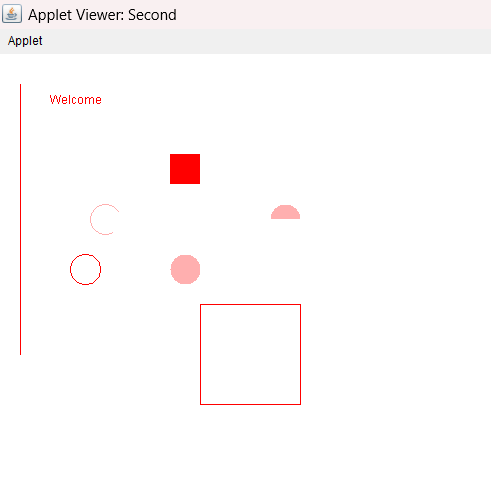
/\*

<applet code="Second" width=300 height=50>

</applet>

\*/

**OUTPUT:**

****

**PRACTICAL 20**

Q: **WAP in Applet to set background and foreground color.**

**CODE:**

import java.awt.\*;

import java.applet.\*;

/\*

<applet code="Third" width=300 height=50>

</applet>

\*/

public class Third extends Applet {

String msg;

public void init() {

    setBackground (Color.cyan);

    setForeground (Color.red);

    msg = "Inside init() --";

}

public void start() {

    msg += " Inside start() --";

}

public void paint (Graphics g) {

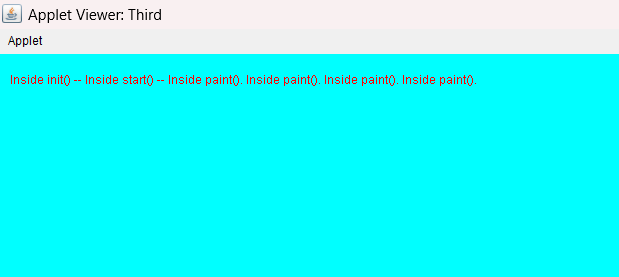
    msg+=" Inside paint().";

    g. drawString (msg, 10, 30);

}

}

**OUTPUT:**

****

**PRACTICAL 21**

Q: **WAP in Applet to Add an image**

**CODE:**

import java.applet.Applet;

import java.awt.Graphics;

import java.awt.Image;

import java.net.URL;

public class ImageApplet extends Applet {

    private Image image;

    public void init() {

        // Get the URL of the image file

        URL imageURL = getClass().getResource("image.jpg");

        // Load the image

        image = getImage(imageURL);

    }

    public void paint(Graphics g) {

        // Draw the image on the applet

        g.drawImage(image, 0, 0, this);

    }

}

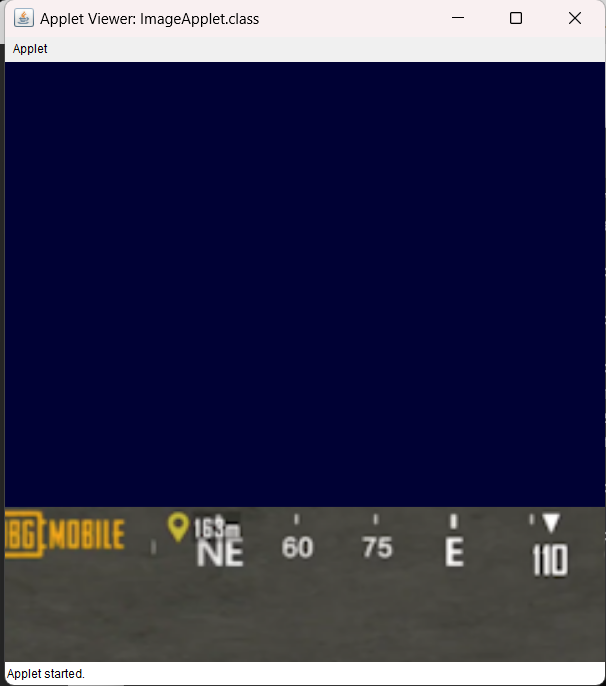
/\*

<applet code ="ImageApplet.class" width="600" height="600">

</applet>

\*/

**OUTPUT:**

****

**PRACTICAL 22**

Q: **WAP to draw a smiley using APPLET**

**CODE:**

import java.applet.\*;

import java.awt.\*;

public class smiley extends Applet {

    public void paint(Graphics g)

    {

        // Oval for face outline

        g.drawOval(80, 70, 150, 150);

        // Ovals for eyes

        // with black color filled

        g.setColor(Color.BLACK);

        g.fillOval(120, 120, 15, 15);

        g.fillOval(170, 120, 15, 15);

        int x[]=  {152,142,163,152};

        int y[]=  {150,169,169,150};

        g.drawPolygon(x, y, 4);

        //g.drawLine (150,130,150,170);

        // Arc for the smile

        g.drawArc(130, 180, 50, 20, 180, 180);

    }

}

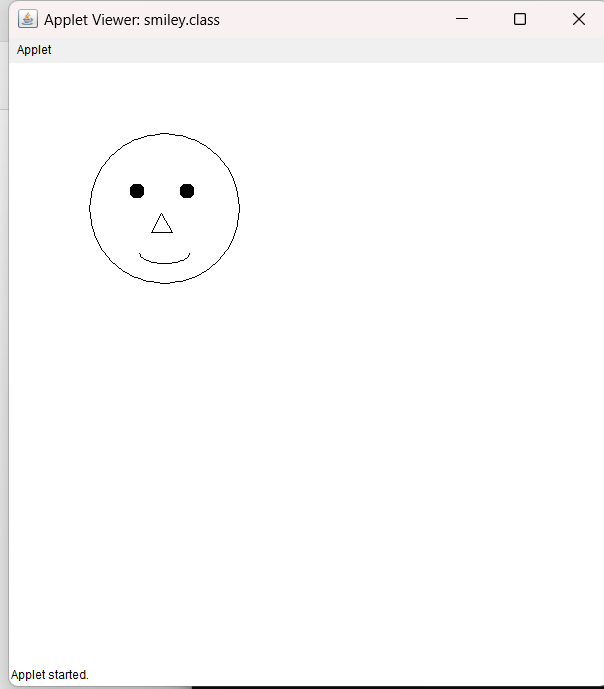
/\*

<applet code ="smiley.class" width="600" height="600">

</applet>

\*/

**OUTPUT:**

****

**PRACTICAL 23**

Q: WAP to create a registration using AWT

**CODE:**

import java.awt.\*;

import java.awt.event.\*;

public class RegistrationForm extends Frame {

    // Label components

    private Label nameLabel;

    private Label rollNoLabel;

    private Label genderLabel;

    private Label subjectLabel;

    private Label collegeLabel;

    private Label remarksLabel;

    // Text field components

    private TextField nameField;

    private TextField rollNoField;

    // Checkbox component for gender

    private CheckboxGroup genderGroup;

    private Checkbox maleCheckbox;

    private Checkbox femaleCheckbox;

    private Checkbox otherCheckbox;

    // Choice component for subject

    private Choice subjectChoice;

    // List component for colleges

    private List collegeList;

    // Text area component for remarks

    private TextArea remarksArea;

    // Button component

    private Button submitButton;

    public RegistrationForm() {

        // Set the layout

        setLayout(new GridBagLayout());

        // Create labels

        nameLabel = new Label("Name:");

        rollNoLabel = new Label("Roll No.:");

        genderLabel = new Label("Gender:");

        subjectLabel = new Label("Subject Choice:");

        collegeLabel = new Label("Colleges:");

        remarksLabel = new Label("Remarks:");

        // Create text fields

        nameField = new TextField(20);

        rollNoField = new TextField(10);

        // Create checkboxes for gender

        genderGroup = new CheckboxGroup();

        maleCheckbox = new Checkbox("Male", genderGroup, true);

        femaleCheckbox = new Checkbox("Female", genderGroup, false);

        otherCheckbox = new Checkbox("Other", genderGroup, false);

        // Create choice for subject

        subjectChoice = new Choice();

        subjectChoice.add("Mathematics");

        subjectChoice.add("Science");

        subjectChoice.add("History");

        subjectChoice.add("English");

        subjectChoice.add("Computer Science");

        // Create list for colleges

        collegeList = new List(3);

        collegeList.add("ABC College");

        collegeList.add("XYZ College");

        collegeList.add("PQR College");

        // Create text area for remarks

        remarksArea = new TextArea(5, 20);

        // Create submit button

        submitButton = new Button("Submit");

        // Add components to the frame using GridBagLayout

        GridBagConstraints constraints = new GridBagConstraints();

        constraints.anchor = GridBagConstraints.WEST;

        constraints.insets = new Insets(5, 5, 5, 5);

        // Add name components

        constraints.gridx = 0;

        constraints.gridy = 0;

        add(nameLabel, constraints);

        constraints.gridx = 1;

        constraints.gridy = 0;

        add(nameField, constraints);

        // Add roll no components

        constraints.gridx = 0;

        constraints.gridy = 1;

        add(rollNoLabel, constraints);

        constraints.gridx = 1;

        constraints.gridy = 1;

        add(rollNoField, constraints);

        // Add gender components

        constraints.gridx = 0;

        constraints.gridy = 2;

        add(genderLabel, constraints);

        Panel genderPanel = new Panel();

        genderPanel.add(maleCheckbox);

        genderPanel.add(femaleCheckbox);

        genderPanel.add(otherCheckbox);

        constraints.gridx = 1;

        constraints.gridy = 2;

        add(genderPanel, constraints);

        // Add subject choice components

        constraints.gridx = 0;

        constraints.gridy = 3;

        add(subjectLabel, constraints);

        constraints.gridx = 1;

        constraints.gridy = 3;

        add(subjectChoice, constraints);

        // Add college components

        constraints.gridx = 0;

        constraints.gridy = 4;

        add(collegeLabel, constraints);

        constraints.gridx = 1;

        constraints.gridy = 4;

        add(collegeList, constraints);

        // Add remarks components

        constraints.gridx = 0;

        constraints.gridy = 5;

        add(remarksLabel, constraints);

        constraints.gridx = 1;

        constraints.gridy = 5;

        add(remarksArea, constraints);

        // Add submit button

        constraints.gridx = 0;

        constraints.gridy = 6;

        constraints.gridwidth = 2;

        constraints.anchor = GridBagConstraints.CENTER;

        add(submitButton, constraints);

        // Add action listener to the button

        submitButton.addActionListener(new ActionListener() {

            public void actionPerformed(ActionEvent e) {

                // Retrieve input values

                String name = nameField.getText();

                String rollNo = rollNoField.getText();

                String gender = genderGroup.getSelectedCheckbox().getLabel();

                String subject = subjectChoice.getSelectedItem();

                String college = collegeList.getSelectedItem();

                String remarks = remarksArea.getText();

                // TODO: Perform registration logic with the input values

                // Clear fields after submission

                nameField.setText("");

                rollNoField.setText("");

                genderGroup.setSelectedCheckbox(maleCheckbox);

                subjectChoice.select(0);

                collegeList.select(0);

                remarksArea.setText("");

            }

        });

        // Set frame properties

        setTitle("Registration Form");

        setSize(400, 400);

        setVisible(true);

    }

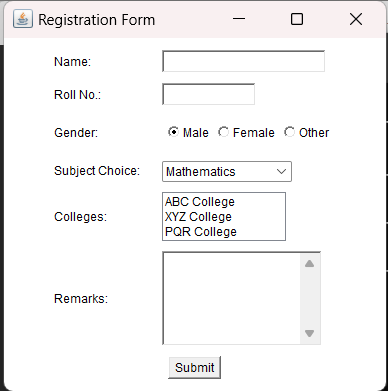
    public static void main(String[] args) {

        new RegistrationForm();

    }

}

**OUTPUT:**

****

**PRACTICAL 23**

Q: WAP to design a calculator layout(using AWT Grid layout).

**CODE:**

import java.awt.\*;

public class MyGridLayout{

    Frame f;

    MyGridLayout(){

        f=new Frame();

        Button b1 =new Button("1");

        Button b2 =new Button("2");

        Button b3 =new Button("3");

        Button b4 =new Button("4");

        Button b5 =new Button("5");

        Button b6 =new Button("6");

        Button b7 =new Button("7");

        Button b8 =new Button("8");

        Button b9 =new Button("9");

        Button b10 =new Button("0");

        Button b11 =new Button("+");

        Button b12 =new Button("\*");

        Button b13 =new Button("%");

        Button b14 =new Button("-");

        Button b15 =new Button("C");

        Button b16 =new Button("clear");

        f.add(b1);f.add(b2);f.add(b3);f.add(b4);f.add(b5);

        f.add(b6);f.add(b7);f.add(b8);f.add(b9);f.add(b10);

        f.add(b11);f.add(b12);f.add(b13);f.add(b14);f.add(b15);f.add(b16);

        f.setLayout(new GridLayout(4,4));

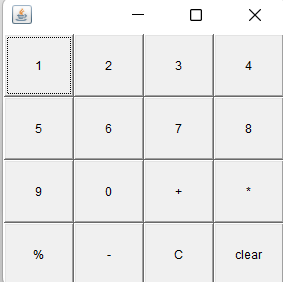
        f.setSize(300,300);

        f.setVisible(true);}

    public static void main(String [] args){

        new MyGridLayout(); }}

**OUTPUT:**



**PRACTICAL 24**

Q: WAP to Java Program to Create 2 Radio Buttons and Display Selected Button Label.

**CODE:**

import java.awt.\*;

import java.awt.event.\*;

public class RadioButtonExample extends Frame {

    private Label resultLabel;

    public RadioButtonExample() {

        setTitle("Radio Button Example");

        setLayout(new GridLayout(0, 1));

        resultLabel = new Label("Selected Button: ");

        add(resultLabel);

        CheckboxGroup checkboxGroup = new CheckboxGroup();

        Checkbox radioBtn1 = new Checkbox("Option 1", checkboxGroup, false);

        radioBtn1.addItemListener(new ItemListener() {

            public void itemStateChanged(ItemEvent e) {

                if (radioBtn1.getState()) {

                    resultLabel.setText("Selected Button: " + radioBtn1.getLabel());

                }

            }

        });

        add(radioBtn1);

        Checkbox radioBtn2 = new Checkbox("Option 2", checkboxGroup, false);

        radioBtn2.addItemListener(new ItemListener() {

            public void itemStateChanged(ItemEvent e) {

                if (radioBtn2.getState()) {

                    resultLabel.setText("Selected Button: " + radioBtn2.getLabel());

                }

            }

        });

        add(radioBtn2);

        pack();

        setLocationRelativeTo(null);

        setVisible(true);

        addWindowListener(new WindowAdapter() {

            public void windowClosing(WindowEvent e) {

                dispose();

            }

        });

    }

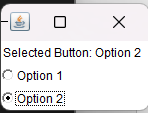
    public static void main(String[] args) {

        new RadioButtonExample();

    }

}

**OUTPUT:**

****

**PRACTICAL 25**

Q: WAP to Create a calculator using Swings

**CODE:**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class CalculatorApp extends JFrame implements ActionListener {

    private JTextField inputField;

    public CalculatorApp() {

        setTitle("Calculator");

        setSize(300, 400);

        setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        // Create the input field

        inputField = new JTextField();

        inputField.setPreferredSize(new Dimension(280, 40));

        inputField.setFont(new Font("Arial", Font.PLAIN, 20));

        inputField.setEditable(false);

        // Create buttons

        String[] buttonLabels = {"7", "8", "9", "/", "4", "5", "6", "\*", "1", "2", "3", "-", "0", ".", "=", "+"};

        JPanel buttonPanel = new JPanel(new GridLayout(4, 4, 5, 5));

        for (String label : buttonLabels) {

            JButton button = new JButton(label);

            button.setFont(new Font("Arial", Font.PLAIN, 20));

            button.addActionListener(this);

            buttonPanel.add(button);

        }

        // Add components to the content pane

        Container container = getContentPane();

        container.setLayout(new BorderLayout());

        container.add(inputField, BorderLayout.NORTH);

        container.add(buttonPanel, BorderLayout.CENTER);

        setVisible(true);

    }

    public void actionPerformed(ActionEvent e) {

        String action = e.getActionCommand();

        String currentText = inputField.getText();

        if (action.equals("=")) {

            try {

                double result = evaluateExpression(currentText);

                inputField.setText(Double.toString(result));

            } catch (ArithmeticException ex) {

                inputField.setText("Error");

            }

        } else if (action.equals("C")) {

            inputField.setText("");

        } else {

            inputField.setText(currentText + action);

        }

    }

    private double evaluateExpression(String expression) {

        try {

            return new Object() {

                int index = -1, ch;

                void nextChar() {

                    ch = (++index < expression.length()) ? expression.charAt(index) : -1;

                }

                boolean isDigit() {

                    return Character.isDigit(ch);

                }

                double parse() {

                    nextChar();

                    double x = parseExpression();

                    if (index < expression.length()) throw new RuntimeException("Unexpected: " + (char)ch);

                    return x;

                }

                double parseExpression() {

                    double x = parseTerm();

                    while (true) {

                        if (eat('+')) x += parseTerm();

                        else if (eat('-')) x -= parseTerm();

                        else return x;

                    }

                }

                double parseTerm() {

                    double x = parseFactor();

                    while (true) {

                        if (eat('\*')) x \*= parseFactor();

                        else if (eat('/')) x /= parseFactor();

                        else return x;

                    }

                }

                double parseFactor() {

                    if (eat('+')) return parseFactor();

                    if (eat('-')) return -parseFactor();

                    double x;

                    int startPos = index;

                    if (eat('(')) {

                        x = parseExpression();

                        eat(')');

                    } else if (isDigit()) {

                        while (isDigit()) nextChar();

                        x = Double.parseDouble(expression.substring(startPos, index));

                    } else {

                        throw new RuntimeException("Unexpected: " + (char)ch);

                    }

                    return x;

                }

                boolean eat(int charToEat) {

                    while (ch == ' ') nextChar();

                    if (ch == charToEat) {

                        nextChar();

                        return true;

                    }

                    return false;

                }

            }.parse();

        } catch (Exception e) {

            throw new ArithmeticException("Invalid expression");

        }

    }

    public static void main(String[] args) {

        SwingUtilities.invokeLater(() -> {

            new CalculatorApp();

        });

    }

}

**OUTPUT:**



**PRACTICAL 26**

Q: WAP to get Hostname IP address and username

**CODE:**

import java.net.InetAddress;

public class SystemInfo {

    public static void main(String[] args) {

        try {

            // Get hostname

            String hostname = InetAddress.getLocalHost().getHostName();

            System.out.println("Hostname: " + hostname);

            // Get IP address

            String ipAddress = InetAddress.getLocalHost().getHostAddress();

            System.out.println("IP Address: " + ipAddress);

            // Get username

            String username = System.getProperty("user.name");

            System.out.println("Username: " + username);

        } catch (Exception e) {

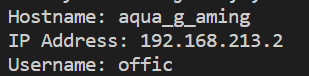
            e.printStackTrace();

        }

    }

}

**OUTPUT:**

****

**PRACTICAL 27**

Q: WAP to implement client side and server side coding

**CODE:**

**Server Side**

import java.io.\*;

import java.net.\*;

public class Server {

    public static void main(String[] args) {

        try {

            // Create a ServerSocket that listens on port 1234

            ServerSocket serverSocket = new ServerSocket(1234);

            System.out.println("Server started. Waiting for client...");

            // Wait for a client to connect

            Socket clientSocket = serverSocket.accept();

            System.out.println("Client connected: " + clientSocket);

            // Create input and output streams for communication with the client

            BufferedReader in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));

            PrintWriter out = new PrintWriter(clientSocket.getOutputStream(), true);

            // Start reading messages from the client and respond

            String message;

            while ((message = in.readLine()) != null) {

                System.out.println("Client: " + message);

                out.println("Server received: " + message);

            }

            // Close the connections

            in.close();

            out.close();

            clientSocket.close();

            serverSocket.close();

        } catch (IOException e) {

            e.printStackTrace(); } }}

**Client Side**

import java.io.\*;

import java.net.\*;

public class Client {

    public static void main(String[] args) {

        try {

            // Create a socket that connects to the server on port 1234

            Socket socket = new Socket("localhost", 1234);

            System.out.println("Connected to server.");

            // Create input and output streams for communication with the server

            BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));

            PrintWriter out = new PrintWriter(socket.getOutputStream(), true);

            // Read user input from the console

            BufferedReader consoleInput = new BufferedReader(new InputStreamReader(System.in));

            String userInput;

            while ((userInput = consoleInput.readLine()) != null) {

                // Send the user input to the server

                out.println(userInput);

                // Receive and print the server's response

                String response = in.readLine();

                System.out.println("Server: " + response);

            }

            // Close the connections

            in.close();

            out.close();

            socket.close();

        } catch (IOException e) {

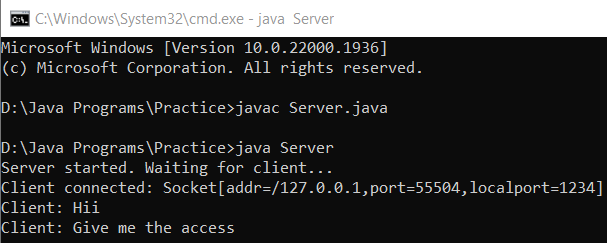
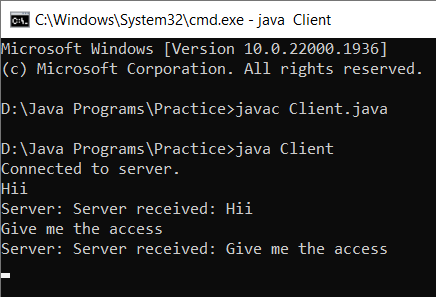
            e.printStackTrace();

        }

    }

}

**OUTPUT:**

**PRACTICAL 28**

Q: WAP to implement JDBC concept by creating a database in XAMP and connect it using Java.

**CODE:**

import java.sql.\*;

public class MysqlCon{

public static void main(String args[]){

try{

Class.forName("com.mysql.jdbc.Driver");

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb","root","");

//here sonoo is database name, root is username and password

Statement stmt=con.createStatement();

ResultSet rs=stmt.executeQuery("select \* from myguests");

while(rs.next())

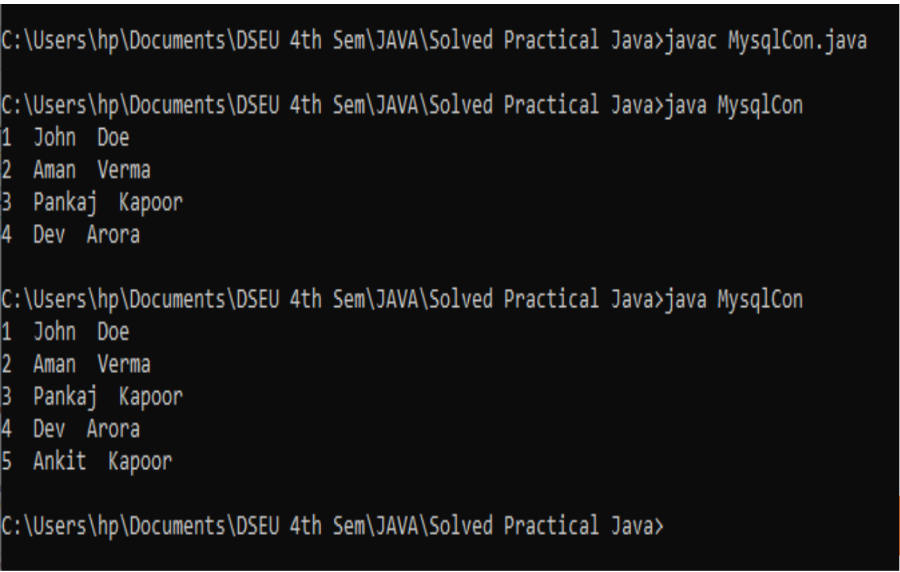
System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getString(3));

con.close();

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

}}

**OUTPUT:**



**PRACTICAL 29 (Project Work)**

Q: Create a puzzle game using Swings through AWT api with event handling.

**CODE:**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.ArrayList;

import java.util.Collections;

import java.util.List;

public class PuzzleGame extends JFrame implements ActionListener {

    private JPanel panel;

    private List<JButton> puzzlePieces;

    private JButton resetButton;

    private int emptyIndex;

    private JButton emptyPiece;

    public PuzzleGame() {

        setTitle("Puzzle Game");

        setSize(400, 400);

        setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        panel = new JPanel(new GridLayout(4, 4));

        puzzlePieces = new ArrayList<>();

        // Create puzzle pieces

        for (int i = 1; i <= 15; i++) {

            JButton button = new JButton(String.valueOf(i));

            button.addActionListener(this);

            puzzlePieces.add(button);

        }

        // Create empty piece

        emptyPiece = new JButton("");

        emptyPiece.setEnabled(false);

        puzzlePieces.add(emptyPiece);

        // Shuffle the puzzle pieces

        Collections.shuffle(puzzlePieces);

        // Add puzzle pieces to the panel

        for (JButton piece : puzzlePieces) {

            panel.add(piece);

        }

        resetButton = new JButton("Reset");

        resetButton.addActionListener(this);

        // Add components to the frame

        add(panel, BorderLayout.CENTER);

        add(resetButton, BorderLayout.SOUTH);

    }

    @Override

    public void actionPerformed(ActionEvent e) {

        JButton source = (JButton) e.getSource();

        if (source == resetButton) {

            // Reset the puzzle by shuffling the pieces

            Collections.shuffle(puzzlePieces);

        } else {

            int sourceIndex = puzzlePieces.indexOf(source);

            if (isAdjacent(sourceIndex, emptyIndex)) {

                // Swap the puzzle piece with the empty piece

                Collections.swap(puzzlePieces, sourceIndex, emptyIndex);

            }

        }

        // Update the panel with the new arrangement of pieces

        panel.removeAll();

        for (JButton piece : puzzlePieces) {

            panel.add(piece);

        }

        panel.revalidate();

        panel.repaint();

        // Update the index of the empty piece

        emptyIndex = puzzlePieces.indexOf(emptyPiece);

    }

    private boolean isAdjacent(int index1, int index2) {

        int row1 = index1 / 4;

        int col1 = index1 % 4;

        int row2 = index2 / 4;

        int col2 = index2 % 4;

        return (Math.abs(row1 - row2) == 1 && col1 == col2) || (Math.abs(col1 - col2) == 1 && row1 == row2);

    }

    public static void main(String[] args) {

        SwingUtilities.invokeLater(new Runnable() {

            @Override

            public void run() {

                PuzzleGame game = new PuzzleGame();

                game.setVisible(true);

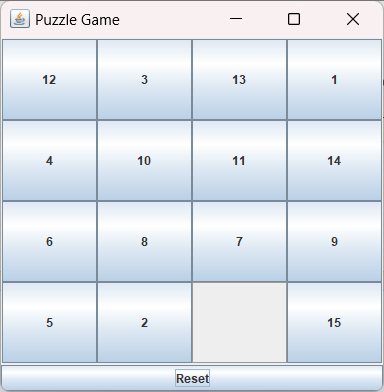
            }

        });

    }

}

**OUTPUT:**

****