

Experiment No. 1

Title: Implementation of class and object

Name: Harshwardhan Mahesh Sawant

Roll No. 50

TY CSE Batch: T3

Q.1.1 Write a program to multiply matrix.

```
import java.util.Scanner;

public class Matrixmul {

    public static void printmatrix(int[][] m, int rowsize, int colsize) {
        for (int i = 0; i < rowsize; i++) {
            for (int j = 0; j < colsize; j++){
                System.out.print(m[i][j] + " ");
            }
            System.out.println();
        }
    }

    public static int[][] multiplymatrix(int[][] A, int[][] B) {
        int i, j, k;
        int row1 = A.length, col1 = A[0].length, row2 = B.length, col2 = B[0].length;
        if (row2 != col1) {
            throw new IllegalArgumentException("Matrix dimension invalid");
        }
        int[][] C = new int[row1][col2];
        for (i = 0; i < row1; i++) {
            for(j = 0; j < col2; j++) {
                for(k = 0; k < row2; k++) {
                    C[i][j] += A[i][k] * B[k][j];
                }
            }
        }
    }
}
```

```

    }
}
return C;
}

public static int[][] inputmatrix(String name) {
    int row, col;

    Scanner sc = new Scanner(System.in);

    System.out.print("Enter the number of rows in " + name + ": ");

    row = sc.nextInt();

    System.out.print("Enter the number of columns in " + name + ": ");

    col = sc.nextInt();

    int[][] matrix = new int[row][col];

    for (int i = 0; i < row; i++) {
        for(int j = 0; j < col; j++) {
            System.out.printf("Enter the element at position [%d][%d]:", i, j);

            matrix[i][j] = sc.nextInt();
        }
    }

    return matrix;
}

public static void main(String[] args) {
    int[][] A = inputmatrix("A");
    int[][] B = inputmatrix("B");
    int[][] C = multiplymatrix(A, B);

    System.out.println("\n Product of the above matrices is: ");

    printmatrix(C, C.length, C[0].length);
}
}

```

Output:

Experiment No. 1

Q.1.2 Write a java program to implement a scanner class.

```
import java.util.Scanner;

public class ScannerImpl {

    public static void main(String[] args){

        Scanner sc = new Scanner(System.in);

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

        int num = sc.nextInt();

        System.out.println("You entered: " + num);

        System.out.print("Enter a string: ");

        String str = sc.next();

        System.out.println("You entered: " + str);

        sc.close();

    }

}
```

Output:

Experiment No. 2

Title: Implementation of inheritance and string function.

Name: Harshwardhan Mahesh Sawant

Roll No. 50

TY CSE Batch: T3

Q.2.1 Write program to implement Inheritance.

```
class Super{
    public void display(){
        System.out.println("Super class");
    }
}

class Sub extends Super{
    public void display(){
        System.out.println("Sub class");
    }
}

class Sub2 extends Super{
    public void display(){
        System.out.println("Sub2 class");
    }
}

class Multilevelsub extends Sub{
    public void display(){
        System.out.println("Multilevel class");
    }
}
```

```

    }
}
class Hierarchicalseb extends Super{
    public void display(){
        System.out.println("Hierarchicalseb class");
    }
}
public class Inheritance{
    public static void main(String[] args){
        Super b = new Super();
        Sub d = new Sub();
        Super b2 = new Sub();
        Sub2 d2 = new Sub2();
        Multilevelsub mid = new Multilevelsub();
        Hierarchicalseb hd = new Hierarchicalseb();
        Super b3 = new Hierarchicalseb();
        Super[]supers ={
            b,d,b2,d2,mid,hd,b3
        };
        for(Super Super : supers ){
            Super.display();
        }
    }
}

```

Output:

Experiment No. 2

Q.2.2 Write a program to Implementation of string function.

```
public class StringFns{  
    public static void main(String[] args){  
        String str = "Hello World!";  
        System.out.println("String: "+str);  
        System.out.println("Length: "+str.length());  
        System.out.println("Character at index 0: "+str.charAt(0));  
        System.out.println("Substring from index 0 to 5: "+str.substring(0,5));  
        System.out.println("Index of 'o': "+str.indexOf('o'));  
        System.out.println("Index of 'o' from index 5 : " + str.indexOf('o', 5));  
        System.out.println("Index of \"|o\": "+str.indexOf("|o"));  
        System.out.println("Index of \"|o\" from index 5 : " + str.indexOf("|o", 5));  
        System.out.println("Last index of 'o': " + str.lastIndexOf('o'));  
        System.out.println("Last index of \"|o\": : " + str.lastIndexOf("|o"));  
        System.out.println("Concatenaion: " +str.concat("How are you?"));  
        System.out.println("Replace 'o' with 'a': " + str.replace('o', 'a'));  
        System.out.println("Replace \"|o\" with \"|a\": " + str.replace("|o", "|a"));  
        System.out.println("Uppercase: " + str.toUpperCase());  
        System.out.println("LowerCase: " + str.toLowerCase());  
        System.out.println("Trim: " + " Hello world!".trim());  
    }  
}
```

Output:

Experiment No. 3

Title: Implementation of Encapsulation.

Name: Harshwardhan Mahesh Sawant

Roll No. 50

TY CSE Batch: T3

Q.1 Write a program to Implement a concept of encapsulation

(Create a class called Employee that includes the three pieces of information as a instance variable, last name, first name, monthly salary, class should have a constructor which initialize the variable which provide the set and get method for each instance variable. If monthly salary is not positive set is 0. Create two Employee objects yearly salary. Then give each employee a 10% raise and display it.)

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

```
public class Employee{
```

```
private String firstName;

private String lastName;

private double monthlySalary;

public Employee(String firstName, String lastName, double monthlySalary) {

    this.firstName = firstName;

    this.lastName = lastName;

    this.monthlySalary = monthlySalary;

}

public static void main(String[] args) {

    Employee employee1 = new Employee("John", "Dee", 1000);

    Employee employee2 = new Employee("Jane", "Doe", 2000);

    System.out.println("Employee 1:" + employee1.getFirstName() + employee1.getLastName());

    System.out.println("Employee 2:" + employee2.getFirstName() + employee2.getLastName());

    System.out.println("Employee 1 yearly salary:" + employee1.getYearlySalary());

    System.out.println("Employee 2 yearly salary:" + employee2.getYearlySalary());

    employee1.raiseSalary(10);

    employee2.raiseSalary(10);

    System.out.println("\nAfter 10% raise:");

    System.out.println("Employee 1 yearly salary:" + employee1.getYearlySalary());

    System.out.println("Employee 2 yearly salary:" + employee2.getYearlySalary());

}

public String getFirstName() {

    return firstName;

}

public void setFirstName(String firstName) {

    this.firstName = firstName;

}

public String getLastName() {

    return lastName;

}
```



```
public void setLastName(String lastName) {  
    this.lastName = lastName;  
}  
public double getMonthlySalary(){  
    return monthlySalary;  
}  
public void setMonthlySalary(double monthlySalary){  
    if(monthlySalary < 0){  
        this.monthlySalary = 0;  
    }else{  
        this.monthlySalary = monthlySalary;  
    }  
}  
public double getYearlySalary(){  
    return monthlySalary * 12;  
}  
public void raiseSalary(double percentage){  
    monthlySalary += monthlySalary * percentage / 100;  
}  
}
```

Output :

Experiment No. 4

Title: Implementation of Overloading and Overriding.

Name: Harshwardhan Mahesh Sawant

Roll No. 50

TY CSE Batch: T3

Q.4.1 Program to illustrate concept of overloading.

```
class Adder{  
    static int add(int a, int b){  
        return a + b;  
    }  
    static int add(int a, int b, int c){  
        return a + b + c;  
    }  
    static double add(double a, double b){  
        return a+b;  
    }  
}  
  
public class Overloading{
```

```
public static void main(String[] args){  
    System.out.println("Sum of 1 and 2:" + Adder.add(1 , 2));  
    System.out.println("Sum of 1.0 and 2.0:" + Adder.add(1.0 , 2.0));  
    System.out.println("Sum of 1.2 and 3:" +Adder.add(1.2 , 3));  
  
    }  
}
```

Output :

Q.4.2. Program for Overriding concept.

```
class OverridingBase{
    public void f(){
        System.out.println("Running OverridingBase.foo()");
    }
    public void g(){
        System.out.println("Running OverridingBase.bar()");
    }
}

public class Overriding extends OverridingBase{
    public static void main(String[] args){
        OverridingBase base = new OverridingBase();
        base.f();
        base.g();

        OverridingBase overriding = new Overriding();
        overriding.f();
        overriding.g();
    }

    public void f(){
        System.out.println("Running Overriding.foo()");
    }

    public void g(){
        System.out.println("Running Overriding.bar()");
    }
}
```

```
}  
}
```

Output :

Experiment No. 5

Title : Impementation of static variables and static methods.

Name : Harshwardhan Mahesh Sawant

Roll No. 50

TY CSE Batch: T3

Q. Write a program to implement Static Variable and Method

(Create class SavingsAccount. Use a static variable annualInterestRate to store the annual interest rate for all account holders. Each object of the class contains a private variable savingsBalance indicating the interest by multiplying the savingsBalance by annualInterestRate divided by 12 this interest should be added to savingsBalance. Provide a static method modifyInterestRate that sets the annualInterestRate to a new value

Write a program to test class SavingsAccount. Instantiate two savingsAccount objects, saver1 and saver2, with balances of Rs.2000.00 and Rs.3000.00, respectively. Set annualInterestRate to 4%, then calculate the monthly interest and print the new balances for both savers. Then set annualInterestRate to 5%, calculate the next month's interest and print the new balances for both savers.)

```
public class SavingAccount{  
    private static double annualIntrestRate=0.04;  
    private double savingsBalance;  
    public SavingAccount(double savingsBalance) {  
        this.savingsBalance=savingsBalance;  
    }  
    public static void modifyIntrestRate(double newIntrestRate) {  
        annualIntrestRate=newIntrestRate;  
    }  
    public static void main(String[] args) {
```

```

        SavingAccount saver1 = new SavingAccount(2000);
        SavingAccount saver2 = new SavingAccount(3000);
        System.out.println("\t\tSaver1\t\tSaver2");
System.out.println("Intrest\t\t"+saver1.claculateMonthlyIntrest()+"\t"+saver2.claculateMonthlyIntrest());
        System.out.println("Balance\t\t"+saver1.savingsBalance+"\t"+saver2.savingsBalance);
        SavingAccount.modifyIntrestRate(0.05);

        System.out.println("\n\n\t\tSaver1\t\tSaver2");

System.out.println("Intrest\t\t"+saver1.claculateMonthlyIntrest()+"\t"+saver2.claculateMonthlyIntrest());
        System.out.println("Balance\t\t"+saver1.savingsBalance+"\t"+saver2.savingsBalance);

    }

    public double claculateMonthlyIntrest(){

        double monthlyIntrest=Math.round(savingsBalance*(annualIntrestRate/12)*100.0)/100.0;
        savingsBalance+=monthlyIntrest;
        return monthlyIntrest;
    }
}

```

Output :

Experiment No. 6

Title : Impelementation of Interface.

Name : Harshwardhan Mahesh Sawant

Roll No. 50

TY CSE Batch: T3

Q. Write a program to implement Interface using extends Keyword

(Create Vehicle Interface with name, maxPassanger, and maxSpeed variable. Create LandVehicle and SeaVehicleInterface from Vehicle interface. LandVehicle has num Wheels variable and drive method. SeaVehicle has displacement variable and launch method. Create Car class from LandVehicle, HoverCraft from LandVehicle and SeaVehicle interface. Also create Ship from SeaVehicle. Provide additional methods in HoverCraft as enterLand and enterSea. Similarly provide other methods for class Car and Ship. Demonstrate all classes in application.)

```
interface Vehicle {
    String name = "";
    int maxPassengers = 0;
    int maxSpeed = 0;
}
interface SeaVehicle extends Vehicle {
    int displacement = 0;
    void launch();
}
interface LandVehicle extends Vehicle {
    int numWheels = 0;
    void drive();
}
class Car implements LandVehicle {
    public int numWheels = 4;
    public int maxPassengers = 5

    public int maxSpeed = 180;
    public String name = "Car";
    public static void main(String[] args) {
        Car car = new Car();
        car.drive();
        System.out.println("\nName\t\t:" + car.name);
        System.out.println("NumWheels\t:" + car.numWheels); System.out.println("MaxPassengers\t:" +
        car.maxPassengers); System.out.println("MaxSpeed\t:" + car.maxSpeed);
```

```

}
@Override
public void drive() {

    System.out.println("* DRIVING A CAR *");

}
}
class Hovercraft implements SeaVehicle, LandVehicle {
    public int displacement=10;
    public int numWheels=-0;
    public int maxPassengers=8;
    public int maxSpeed=70;
    public String name="Hovercraft";
    public static void main(String[] args) {
        Hovercraft hovercraft = new Hovercraft();
        hovercraft.enterland();
        hovercraft.drive();
        hovercraft.enterSea();
        hovercraft.launch();
        System.out.println("\nName\t\t\t:" + hovercraft.name);
        System.out.println("Displacement\t\t:" + hovercraft.displacement);
        System.out.println("NumWheels\t\t:" + hovercraft.numWheels);

        System.out.println("MaxPassengers\t\t:" +
hovercraft.maxPassengers);          System.out.println("MaxSpeed\t\t:" + hovercraft.maxSpeed);
    }
    @Override
    public void drive() {

        System.out.println("* DRIVING A HOVERCEAFT *");

    }
    @Override
    public void launch() {
        System.out.println("Launching a hovercraft");
    }
    public void enterland() {
        System.out.println("Entering land");
    }
    public void enterSea() {
        System.out.println("\nEntering sea");
    }
}
class Ship implements SeaVehicle {
    public int displacement = 10;
    public int maxPassengers = 25;
    public int maxSpeed = 35;
    public String name = "Ship";
    public static void main(String[] args) {

```



```

        Ship ship = new Ship();
        ship.launch();
        System.out.println("\nName\t\t:" + ship.name);
        System.out.println("Displacement\t:" + ship.displacement);
        System.out.println("MaxPassengers\t:" + ship.maxPassengers);
        System.out.println("MaxSpeed\t:" + ship.maxSpeed);

    }
    @Override
    public void launch() {

        System.out.println("* LAUNCHING A SHIP *");

    }
}

```

Output:

Experiment No. 7

Title : Impelementation of Abstract class.

Name : Harshwardhan Mahesh Sawant

Roll No. 50

TY CSE Batch: T3

Q . Write a program to implement Abstract class with example

(Create abstract class Shape which has instance variables side, area and perimeter And methods calculateArea(), calculatePerimeter() as abstract methods and display() as concrete method. Write subclasses which extend Shapeclass like Triangle, Rectangle, Circle, Cube and Square and override .)

```

package Shapes;
public abstract class Shape {
    protected String name;
    protected double side; protected
    double area; protected double
    perimeter;
    public Shape(double side) {
        this.side = side;
    }
    public static void main(String[] args) { Shape[]
        shapes = new Shape[5]; shapes[0] = new
        Shapes.Triangle(5, 10); shapes[1] = new
        Shapes.Rectangle(5, 10); shapes[2] = new
        Shapes.Circle(5); shapes[3] = new
        Shapes.Cube(5); shapes[4] = new
        Shapes.Square(5);
        System.out.printf("%-10s %-10s %-10s", "Name", "Area", "Perimeter");

        System.out.print("\n-");

        for (Shape shape : shapes) {
            shape.calculateArea();
            shape.calculatePerimeter()
            shape.display();
        }
    }

    public abstract void calculateArea();
    public abstract void calculatePerimeter(); public
    void display() {
        System.out.printf("\n%-10s %-10.2f %-10.2f", name, area, perimeter);
    }

}

package Shapes;

public class Circle extends Shape {
    public Circle(double side) {
        super(side); name =

```

```

        "Circle";
    }

    @Override
    public void calculateArea() {
        area = Math.PI * Math.pow(side,2); }
    @Override
    public void calculatePerimeter() {
        perimeter = 2 *Math.PI* side;
    }
}

```

```

package Shapes;

class Cube extends Shape { public
    Cube(double side) {
        super(side); name
        = "Cube";
    }

    @Override
    public void calculateArea() {area
        = 6*side *side;
    }

    @Override
    public void calculatePerimeter() {
        perimeter = 12 * side;
    }
}

```

```

package Shapes;
class Rectangle extends Shape {
    private final double length;

    public Rectangle(double side, double length) {
        super(side);

        this.length = length;
    }

    @Override
    public void calculateArea() {area
        = side * length;
    }
}

```

```

    } }
    @Override
    public void calculatePerimeter() {
        perimeter = 2 * (side + length);
    }

```

```

package Shapes;
class Square extends Shape {

    public Square(double side) {
        super(side);
        name = "Square";
    }

    @Override
    public void calculateArea() {area
        = side * side;
    }
    @Override
    public void calculatePerimeter() {
        perimeter = 4 * side;

    } }

```

```

package Shapes;
class Triangle extends Shape {
    private final double height;

    public Triangle(double side, double height) {
        super(side);
        this.height = height;
        name = "Triangle";

    } }
    @Override
    public void calculateArea() {area
        = 0.5 * side * height;
    }
    @Override
    public void calculatePerimeter() {
        perimeter = 3 * side;
    }

}

```

Output :

Experiment No. 8

Title : Impelementation of Package.

Name : Harshwardhan Mahesh Sawant

Roll No. 50

TY CSE Batch: T3

Write a program to implement concept of package. (Develop a mathematical package for statistical operations Program: like mean, mode, median Average . Standard Deviation . Create a subPackage – convert. In convert package provide class to convert decimal to octal, binary, hex and vice- versa .)

```
package Mathops;
```

```
import static Mathops.Conversions.Conversions.fromDecimalToBase;public class Arithmetic {
```

```

public static void main(String[] args) {
    int[] numbers = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

    double mean = mean(numbers); double
    median = median(numbers); double mode
    = mode(numbers); double average =
    average(numbers);
    double standardDeviation = standardDeviation(numbers);

    System.out.printf("%-20s %-20s", "Statistic", "Value");

    System.out.print("\n
                                ");

    System.out.printf("\n%-20s %-20.2f", "Mean", mean); System.out.printf("\n%-20s %-20.2f",
    "Median", median); System.out.printf("\n%-20s %-20.2f", "Mode", mode);
    System.out.printf("\n%-20s %-20.2f", "Average", average); System.out.printf("\n%-20s %-20.2f",
    "Standard Deviation",

    standardDeviation);

    int decimal = 24;
    String binary = fromDecimalToBase(decimal,2);String
    octal = fromDecimalToBase(decimal,8);

    String hexadecimal = fromDecimalToBase(decimal,16);
    System.out.println("\n\nConversions");

    System.out.printf("%-10s %-10s %-10s %-10s", "Decimal", "Binary", "Octal",
    "Hexadecimal");
    System.out.print("\n-
    System.out.printf("\n%-10s %-10s %-10s %-10s", decimal, binary, octal,
    hexadecimal);
    }

    public static double mean(int[] numbers) {int
    sum = 0;
    for (int number : numbers) {sum
    += number;

    }
    return (double) sum / numbers.length;
    }

    public static double median(int[] numbers) {int
    middle = numbers.length / 2;
    if (numbers.length % 2 == 1) {
    return numbers[middle];
    }
    }

```

```

        } else {
            return (numbers[middle - 1] + numbers[middle]) / 2.0; }
    }
    public static double mode(int[] numbers) {int
        maxValue = 0;

        } int maxCount = 0;
        for (int number : numbers) {int
            count = 0;
            for (int i : numbers) {
                if (i == number) ++count;
            }
            if (count > maxCount) {
                maxCount = count;
                maxValue = number;
            }
        }
        return maxValue;
    }

    public static double average(int[] numbers)
    {int sum = 0;
        for (int number :
            numbers) {sum +=
                number;

    }

    }
        return (double) sum / numbers.length;
    }
    public static double standardDeviation(int[] numbers) {double    mean = mean(numbers);
        double temp = 0;
        for (double number : numbers) {
            temp += (number - mean) * (number - mean);
        }
        return Math.sqrt(temp / numbers.length);
    }

}

```

Output :

Experiment No. 9

Title : Impelementation of Multiple Catch Block.

Name : Harshwardhan Mahesh Sawant

Roll No. 50

TY CSE Batch: T3

Q. Write a program to implement Multiple Catch Block.

```
public class MultipleCatch{  
    public static void main(String[] args){  
        int [] numbers = {1,2,3,4,5,6,7,8,9,10};  
        try{  
            System.out.println(numbers[10]);  
        }  
        catch(ArrayIndexOutOfBoundsException e){  
            System.out.println("Array index out of bounds.");  
            try{  
                int number = 30/0;  
            }catch(ArithmeticException e2){  
                System.out.println("Cannot divide by zero.");  
            }  
        }  
        catch(Exception e){
```



```

        System.out.println("Exception caught.");
    }
}
}

```

OUTPUT:

Experiment No. 10

Title : Impelementation of Abstract class.

Name: Harshwardhan Mahesh Sawant

Roll No. 50

TY CSE Batch: T3

Q. Write a program to implement user defined exception.

(Develop a Bankaccount class which should contain all methods of bank.i.e.balanceenquiry(),withdraw(),transfer() and deposit().You create at least two objects of Bankaccount using array and do all operations mentioned above.)

```

public class BankAccount {
    private final String password;
    private double balance;
    public BankAccount(String name, String password, double balance) {
        this.password = password;
        this.balance = balance;
    } public static void main(String[]args) {
        BankAccount account1 = new BankAccount("John","1234",1000);
        BankAccount account2 = new BankAccount("Jane","1234",1000);
        try {
            account1.withdraw(500,"1234");
            account1.balanceEnquiry();
            account1.withdraw(5000,"1234");
            account1.balanceEnquiry();
            account1.withdraw(5000,"9874");
            account1.balanceEnquiry();
        } catch (LowBalanceException | NegativeNumberEception | passwordMismatchException e) {
            System.out.println(e.getMessage());
        } try {

```

```

        account1.deposit(-500);
        account1.balanceEnquiry();
    } catch(NegativeNumberException e) {
        System.out.println(e.getMessage());
    }
    try {
        account1.transfer(account2, 500, "1234");
        account1.balanceEnquiry();
        account2.balanceEnquiry();

        } catch(LowBalanceException | NegativeNumberException |
passwordMismatchException e) {
            System.out.println(e.getMessage());
        }
    }

    public void balanceEnquiry() {
        System.out.println("Balance: " + balance);
    }

    public void withdraw(double amount, String password) throws LowBalanceException,
    NegativeNumberException, passwordMismatchException
    {
        if(amount < 0) {
            throw new NegativeNumberException("Amount cannot be negative");
        }
        if(!this.password.equals(password)) {
            throw new passwordMismatchException("Password mismatch");
        }
        if(balance - amount < 0) {
            throw new LowBalanceException("Low balance");
        }
        balance -= amount;
    }
    public void deposit(double amount) throws NegativeNumberException {
        if (amount < 0) {
            throw new NegativeNumberException("Negative number");
        }
    }

    balance += amount;
}

    public void transfer(BankAccount other, double amount, String password)throws LowBalanceException,
    NegativeNumberException, passwordMismatchException {

        withdraw(amount,password);
        other.deposit(amount);
    }
}

class LowBalanceException extends Exception {

```

```

        public LowBalanceException(String message) {
            super(message);
        }
    }
    class NegativeNumberException extends Exception {
    public NegativeNumberException(String message) {
    super(message);
        }
    }
    class passwordMismatchException extends Exception {
    public passwordMismatchException(String message) {
    super(message);
        }
    }
}

```

Output :

EXPERIMENT NO:11

Title: Implementation of file properties..

Name: Harshwardhan Mahesh Sawant

Roll No: 50TY CSE: Batch T3

Q – Write a program to implement file properties.

()Take file name as input to your program through command line, if file exists then open and display contents of the file. After displaying contents of file ask user – 1.do you want to add the data at the end of file or 2.replace specified text in file by other text. Based on user's response, then accept data from user and append it to file. If file does not exists then create a fresh new-file and store user data into it. Also. User should type exit on new line to stop the program. Do this program using Character stream classes.//)

Program :-

```
package inheritanceEx;
```

```
import java.io.File;
import
java.io.FileWriter;
import
java.util.Scanner;
```

```
public class FileProps {
public static void main(String[] args) {
Scanner scanner = new
Scanner(System.in);
System.out.print("Enter file path: ");
String filePath =
scanner.nextLine(); File file =
new File(filePath);
```

```
if (file.exists()) {
System.out.println("File
contents: ");
try (Scanner fileScanner = new Scanner(file))
{ while (fileScanner.hasNextLine()) {
System.out.println(fileScanner.nextLine());
}
} catch (Exception
e) {
e.printStackTrace(
);
}
```

```
System.out.print("Do you want to update or overwrite the file? (update/
overwrite):
```

```
");
String choice = scanner.nextLine();
```

```

if (choice.equals("update")) {
    System.out.print("Enter new content:
"); String newContent =
scanner.nextLine();
    try (FileWriter fileWriter = new FileWriter(file,
true)) { fileWriter.write(newContent);
    } catch (Exception
e) {
        e.printStackTrace(
);
    }
}
else if (choice.equals("overwrite")) {
    System.out.print("Enter new content:
"); String newContent =
scanner.nextLine();
    try (FileWriter fileWriter = new
FileWriter(file)) {
        fileWriter.write(newContent);
    } catch (Exception
e) {
        e.printStackTrace(
);
    }
} else
{

    System.out.println("Invalid choice");
}
else
{

    System.out.println("File does not exist");
}
}
}
}

```

Output :-

EXPERIMENT NO:12

Title: Implementation of swing GUI based calculator

Name: Harshwardhan Mahesh Sawant

Roll No: 50TY CSE: Batch T3

Q – Write a program to implement Swing GUI based calculator. Program :-

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

public class Calculator {
    public static void main(String[] args) {
        /* Frame */
        JFrame frame = new JFrame("Calculator");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(360, 500);

        /* Screen */
        JTextField screen = new JTextField(10);
        screen.setFont(new Font("Monospace", Font.PLAIN,
        26));
        screen.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));
        screen.setEditable(false);
        screen.setHorizontalAlignment(JTextField.RIGHT);
        screen.setText("0");
        frame.add(screen, BorderLayout.NORTH);

        /* Keypad */
```

```

JPanel keypad = new JPanel(); keypad.setLayout(new GridLayout(5, 4));
String[] symbols = {"C", "^", "%", "/", "7", "8", "9", "x", "4", "5", "6", "-", "1",
"2", "3", "+", "", "0", ".", "="};

```

```

for (int i = 0; i < symbols.length; i++) {

```

```

    JButton button = new JButton(symbols[i]);
    button.setFont(new Font("Monospace", Font.PLAIN,
20)); int finalI = i;

```

```

    if (isNumber(symbols[i])) { button.addActionListener(e -> {
        if (screen.getText().equals("0")) { screen.setText(symbols[finalI]);
    } else {
        screen.setText(screen.getText() + symbols[finalI]);
    }
    });
}

```

```

if (isOperator(symbols[i])) {
    button.addActionListener(e -> {

```

```

        if (isOperator(screen.getText().substring(screen.getText().length () -
1)))

```

```

screen.setText(screen.getText().substring(0, screen.getText().
length() - 1) + symbols[finalI]);

```

```

    } else {
        screen.setText(screen.getText() + symbols[finalI]);
    }

```

```

    });

```

```

}

```

```

if (symbols[i].equals("C")) {
    button.addActionListener(e -> {
        screen.setText("0");
    });
}

```

```

if (symbols[i].equals("=")) {
    button.addActionListener(e -> {
        screen.setText(calculate(screen.getTe
xt()));
    });
}

```

```

if (symbols[i].equals(".")) {
    button.addActionListener(e -> {
        if (!screen.getText().contains(".")) {
            screen.setText(screen.getText() + symbols[finalI]);
        }
    });
}

```

```

if (symbols[i].equals("")) {

    button.setEnabled(false);
}

```

```

    keypad.add(button);
}

```

```

frame.add(keypad,
BorderLayout.CENTER);
frame.setVisible(true);
}

```

```

private static String calculate(String text) {
    String[] numbers =
text.split("[+×/^\%]"); String[] operators
= text.split("[0-9.]+");

```

```

    double result =
Double.parseDouble(numbers[0]); for (int i =
1; i < numbers.length; i++) {

```



```

        switch (operators[i]) {
            case "+" -> result += Double.parseDouble(numbers[i]); case "-" -> result
            -= Double.parseDouble(numbers[i]);
            case "×" -> result *= Double.parseDouble(numbers[i]); case "/" -> result
            /= Double.parseDouble(numbers[i]);
            case "^" -> result = Math.pow(result,
            Double.parseDouble(numbers[i])); case "%" -> result %=
            Double.parseDouble(numbers[i]);
        }
    }

    return result % 1 == 0 ? String.valueOf((int) result) : String.valueOf(result);
}

private static boolean isNumber(String str) { return str.matches("[0-9]+");
}
private static boolean isOperator(String s) { return s.matches("[+×/%^.-]");
}
}

```

Output :-

EXPERIMENT NO:13

Title: Implementation of Chat Application using TCP

Name: Harshwardhan Mahesh Sawant

Roll No: 50TY CSE: Batch T3

Q –program to create a chat application using TCP/UDP.

Program :-

Server :-

```

import java.net.*;
import java.io.*;

public class ChatServer
{
    public static void main(String[] args) throws Exception
    {
        ServerSocket ss=new
        ServerSocket(2000); Socket
        sk=ss.accept();
        BufferedReader cin=new BufferedReader(new
        InputStreamReader(sk.getInputStream()));
        PrintStream cout=new PrintStream(sk.getOutputStream());
        BufferedReader stdin = new BufferedReader(new
        InputStreamReader(System.in));
        String s;

        while(true)
        {
            s=cin.readLine();
            if(s.equalsIgnoreCase("END"))
            {
                cout.println("BYE
                "); break;
            }
            System.out.print("Client:"+s+"\n");

            System.out.print("Server :
            "); s= stdin.readLine();
            cout.println(s);
        }ss.close();
        sk.close();
        cin.close();
        cout.close();
        stdin.close();
    }
}

```

Client :-

```
import java.net.*;
import java.io.*;

public class ChatClient {
    public static void main(String args[]) throws Exception {
        Socket sk = new Socket("127.0.0.1", 2000);

        BufferedReader sin = new BufferedReader(new
InputStreamReader(sk.getInputStream()));
        PrintStream sout = new PrintStream(sk.getOutputStream());

        BufferedReader stdin = new BufferedReader(new
InputStreamReader(System.in));
        String s;

        while (true) {
            System.out.print("Client
:"); s = stdin.readLine();
            sout.println(s);
            s = sin.readLine();
            System.out.print("Server : " + s +
"\n"); if
(s.equalsIgnoreCase("BYE"))
                break;

        }

        sk.close();
        sin.close();
        sout.close ();
        stdin.clos
e();
    }
}
```

Output :-

EXPERIMENT NO:14

Title: Implementation of Database Handling Program

Name: Harshwardhan Mahesh Sawant

Roll No: 50TY CSE: Batch T3

Q – Write a program for handling databases.

(1. Write a java program that connects to database using jdbc & insert values in it.

2. .Write a java program that connects to database using jdbc.& read values from table.)

Program :-

```
import java.sql.*;
public class Jdbc_Insert {
    private static Statement statement;

    public static void main(String[] args) throws SQLException {
        init();
        statement.execute("create table students(stud_id integer,stud_name
varchar(20),stud_address varchar(30))");
        statement.execute("insert into students
values(001,'Shubham','Kaneri')"); statement.execute("insert into
students values(002,'Prathamesh','Sadoli')"); statement.execute("insert
into students values(003,'Gaurav','Vidani')");
    }
}
```

```

private static void init() {
    String url =
        "jdbc:mysql://localhost:3306/dbname"; String
        username = "root";
        String password = "Pranav4019";

    try {
        Connection con = DriverManager.getConnection(url, username,
            password);
        statement = con.createStatement();
    } catch (SQLException e) {
        System.out.println("Connection Failed");
        e.printStackTrace();
    }

}

import java.sql.*;

public class Jdbc_Read {
    private static Statement statement;

    public static void main(String[] args) throws SQLException {
        init();

        ResultSet rs = statement.executeQuery("SELECT * FROM students");

        System.out.printf("%-5s %-12s %-10s", "ID", "Name",
            "address"); System.out.print("\n-____");

        while (rs.next()) {
            System.out.printf("\n%-5s %-12s %-10s", rs.getInt("stud_id"),
                rs.getString("stud_name"), rs.getString("stud_address"));
        }
    }
}

```

```

    }

    private static void init() {
        String url =
            "jdbc:mysql://localhost:3306/dbname"; String
            username = "root";
            String password = "Pranav4019";

        try {
            Connection con = DriverManager.getConnection(url, username,
                password);
            statement = con.createStatement();
        } catch (SQLException e) {
            System.out.println("Connection Failed");
            e.printStackTrace();
        }
    }
}

```

Output :-

EXPERIMENT NO:15

Title: Write a program for student registration and
 Login Name: Harshwardhan Mahesh Sawant
 Roll No: 50TY CSE: Batch T3

Q – write a program for student registration and login using GUI based standard. And store The data to the database.

Program :-

```
import javax.swing.*;
import java.awt.*;
import java.sql.Connection;
import
java.sql.DriverManager;
import
java.sql.SQLException;
import java.sql.Statement;

public class StudentForm {
private static final String[] formLabels = {"Name", "Class", "Phone No", "Email"};
    private static Statement statement;
    public JTextField[] formFields = new JTextField[formLabels.length];
    StudentForm() {
        try {
            String url =
                "jdbc:mysql://localhost:3306/db_name"; String
            username = "username";
            String password = "password";
            Connection con = DriverManager.getConnection(url, username,
                password);
            statement = con.createStatement();
        } catch (SQLException e) {
            System.out.println("Connection Failed");
        }
        JFrame frame = new JFrame("Student Form");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(500, 250);

        JPanel panel = new JPanel(); panel.setLayout(new
            GridLayout(0, 2));
```

```

for (int i = 0; i <
    formLabels.length; i++) {
    JLabel label = new JLabel(formLabels[i]);
    JTextField textField = new JTextField(20);
    formFields[i] = textField; panel.add(label);
    panel.add(textField);
}
JButton register = new
JButton("Register"); panel.add(register);

```

```

JButton login = new JButton("Login");
panel.add(login);

```

```

register.addActionListener( e -> { try {
    addUserToDatabase();
    JOptionPane.showMessageDialog(frame, "User registered successfully");
} catch (SQLException ex) {
    throw new RuntimeException(ex);
}
});

```

```

login.addActionListener( e -
> { try {
    boolean exists =
    loginUser(); if (exists)
    {
        JOptionPane.showMessageDialog(frame, "Login Successful");
    }
    else {
        JOptionPane.showMessageDialog(frame, "Login Failed");
    }
} catch (SQLException ex) {
    throw new RuntimeException(ex);
}
});

```



```

        frame.add(panel);
        frame.setVisible(true);
    }

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

    private void addUserToDatabase() throws SQLException {
        String name = formFields[0].getText();
        String className = formFields[1].getText();
        String phoneNo = formFields[2].getText();
        String email = formFields[3].getText();

        String query = String.format("INSERT INTO students (name, class, phone, email) VALUES ('%s', '%s', '%s', '%s')", name, className, phoneNo, email);

        statement.executeUpdate(query);
    }

    private boolean loginUser() throws SQLException {
        String email = formFields[3].getText();

        String query = String.format("SELECT * FROM students WHERE email = '%s'", email);

        return statement.executeQuery(query).next();
    }
}

```

Output :-