
Course Outcome 3**Experiment 14****Date: 05.04.2024****Method Overloading****Aim:**

Write a java program to calculate the area of different shapes namely circle, rectangle and triangle using the concept of method overloading.

Program

```
import java.util.Scanner;

public class ShapeArea {

    public void area(int side) {
        System.out.println("Area of square: " + side * side);
    }

    public void area(int length, int width) {
        System.out.println("Area of rectangle: " + length * width);
    }

    public void area(double radius) {
        System.out.println("Area of circle: " + 3.14 * radius * radius);
    }

    public void area(double base, double height) {
        System.out.println("Area of triangle: " + 0.5 * base * height);
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        ShapeArea shape = new ShapeArea();

        System.out.print("Enter the side of the square: ");
        int side = scanner.nextInt();
        shape.area(side);

        System.out.print("Enter the length and width of the rectangle: ");
        int length = scanner.nextInt();
        int width = scanner.nextInt();
        shape.area(length, width);
    }
}
```

```
System.out.print("Enter the radius of the circle: ");  
double radius = scanner.nextDouble();  
shape.area(radius);  
System.out.print("Enter the base and height of the triangle: ");  
double base = scanner.nextDouble();  
double height = scanner.nextDouble();  
shape.area(base, height);  
scanner.close();  
}  
}
```

Output

```
Enter the side of the square: 3  
Area of square: 9  
Enter the length and width of the rectangle: 3 5  
Area of rectangle: 15  
Enter the radius of the circle: 3  
Area of circle: 28.2599  
Enter the base and height of the triangle: 6 8  
Area of triangle: 24.0
```

Experiment 15**Date: 05.04.2024****Constructor - 1****Aim:**

Create a class 'Employee' with data members Empid, Name, Salary, Address and constructors to initialize the data members. Create another class 'Teacher' that inherit the properties of class employee and contain its own data members department, Subjects taught and constructors to initialize these data members and also include display function to display all the data members. Use array of objects to display details of N teachers.

Program

```
import java.util.Scanner;

class Employee {
    int empId;
    String name;
    double salary;
    String address;

    public Employee(int empId, String name, double salary, String address) {
        this.empId = empId;
        this.name = name;
        this.salary = salary;
        this.address = address;
    }
}

class Teacher extends Employee {
    String department;
    String subjectsTaught;

    public Teacher(int empId, String name, double salary, String address, String department,
        String subjectsTaught) {
        super(empId, name, salary, address);
        this.department = department;
        this.subjectsTaught = subjectsTaught;
    }

    public void display() {
        System.out.println("Employee ID: " + empId);
```

```
System.out.println("Name: " + name);
System.out.println("Salary: " + salary);
System.out.println("Address: " + address);
System.out.println("Department: " + department);
System.out.println("Subjects Taught: " + subjectsTaught);
} }

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of teachers: ");
        int numTeachers = scanner.nextInt();
        Teacher[] teachers = new Teacher[numTeachers];
        for (int i = 0; i < numTeachers; i++) {
            System.out.println("Enter details for Teacher " + (i + 1) + ":");
            System.out.print("Employee ID: ");
            int empId = scanner.nextInt();
            scanner.nextLine();
            System.out.print("Name: ");
            String name = scanner.nextLine();
            System.out.print("Salary: ");
            double salary = scanner.nextDouble();
            scanner.nextLine();
            System.out.print("Address: ");
            String address = scanner.nextLine();
            System.out.print("Department: ");
            String department = scanner.nextLine();
            System.out.print("Subjects Taught: ");
            String subjectsTaught = scanner.nextLine();
            teachers[i] = new Teacher(empId, name, salary, address, department, subjectsTaught);
        }
        System.out.println("\nDetails of all teachers:");
```

```
for (int i = 0; i < numTeachers; i++) {  
    System.out.println("\nTeacher " + (i + 1) + ":");  
    teachers[i].display();  
    scanner.close();  
} }
```

Output

Enter the number of teachers: 3

Enter details for Teacher 1:

Employee ID: 23

Name: Anu

Salary: 54000

Address: XYZ(H),Peruva,Kottayam

Department: CE

Subjects Taught: Maths

Enter details for Teacher 2:

Employee ID: 25

Name: Akhila

Salary: 60000

Address: ABC(H),Keezhoor,Kottayam

Department: MCA

Subjects Taught: AI

Enter details for Teacher 3:

Employee ID: 24

Name: Binu

Salary: 75000

Address: MNO(H),Puthencurz,Ernakulam

Department: BCA

Subjects Taught: Java

Details of all teachers:

Teacher 1:

Employee ID: 23

Name: Anu

Salary: 54000.0

Address: XYZ(H) Peruva Kottayam

Department: CE

Subjects Taught: Maths

Teacher 2:

Employee ID: 25

Name: Akhila

Salary: 60000.0

Address: ABC(H),Keezhoor,Kottayam

Department: MCA

Subjects Taught: AI

Teacher 3:

Employee ID: 24

Name: Binu

Salary: 75000.0

Address: MNO(H),Puthencurz,Ernakulam

Department: BCA

Subjects Taught: Java

Experiment 16**Date: 05.04.2024****Constructor - 2****Aim:**

Create a class 'Person' with data members Name, Gender, Address, Age and a constructor to initialize the data members and another class 'Employee' that inherits the properties of class Person and also contains its own data members like Empid, Company_name, Qualification, Salary and its own constructor. Create another class 'Teacher' that inherits the properties of class Employee and contains its own data members like Subject, Department, Teacherid and also contain constructors and methods to display the data members. Use array of objects to display details of N teachers.

Program

```
import java.util.Scanner;

class Person {
    String name;
    String gender;
    String address;
    int age;

    public Person(String name, String gender, String address, int age) {
        this.name = name;
        this.gender = gender;
        this.address = address;
        this.age = age;
    }
}

class Employee extends Person {
    int empId;
    String company_name;
    String qualification;
    double salary;

    public Employee(String name, String gender, String address, int age, int empId, String company_name, String qualification, double salary) {
        super(name, gender, address, age);
        this.empId = empId;
    }
}
```

```
this.company_name = company_name;
this.qualification = qualification;
this.salary = salary;
}
}
class Teacher extends Employee {
String subject;
String department;
int teacherId;
public Teacher(String name, String gender, String address, int age, int empId, String
company_name, String qualification, double salary,
String subject, String department, int teacherId) {
super(name, gender, address, age, empId, company_name, qualification, salary);
this.subject = subject;

this.department = department;
this.teacherId = teacherId;
}
public void display() {
System.out.println("Name: " + name);
System.out.println("Gender: " + gender);
System.out.println("Address: " + address);
System.out.println("Age: " + age);
System.out.println("Employee ID: " + empId);
System.out.println("Company Name: " + company_name);
System.out.println("Qualification: " + qualification);
System.out.println("Salary: " + salary);
System.out.println("Subject: " + subject);
System.out.println("Department: " + department);
System.out.println("Teacher ID: " + teacherId);
}
```



```
}  
  
public class PersonMain1 {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        System.out.print("Enter the number of teachers: ");  
        int numTeachers = scanner.nextInt();  
        Teacher[] teachers = new Teacher[numTeachers];  
        for (int i = 0; i < numTeachers; i++) {  
            System.out.println("Enter details for Teacher " + (i + 1) + ":");  
            System.out.print("Name: ");  
            String name = scanner.next();  
            System.out.print("Gender: ");  
            String gender = scanner.next();  
            System.out.print("Address: ");  
            String address = scanner.next();  
            System.out.print("Age: ");  
            int age = scanner.nextInt();  
            System.out.print("Employee ID: ");  
            int empId = scanner.nextInt();  
            System.out.print("Company Name: ");  
            String companyName = scanner.next();  
            System.out.print("Qualification: ");  
            String qualification = scanner.next();  
            System.out.print("Salary: ");  
            double salary = scanner.nextDouble();  
            System.out.print("Subject: ");  
            String subject = scanner.next();  
            System.out.print("Department: ");  
            String department = scanner.next();  
            System.out.print("Teacher ID: ");  
            int teacherId = scanner.nextInt();
```

```
teachers[i] = new Teacher(name, gender, address, age, empId, companyName,
qualification, salary, subject, department, teacherId);
}
System.out.println("\nDetails of all teachers:");
for (int i = 0; i < numTeachers; i++) {
System.out.println("\nTeacher " + (i + 1) + ":");
teachers[i].display();
}
scanner.close();
}}
```

Output

Enter the number of teachers: 2

Enter details for Teacher 1:

Name: Akhila

Gender: Female

Address: ABC(H),Keezhoor,Kottayam

Age: 24

Employee ID: 2244

Company Name: TCS

Qualification: MCA

Salary: 45000

Subject: Maths

Department: BCA

Teacher ID: 34

Enter details for Teacher 2:

Name: Anu

Gender: Female

Address: MNO(H),Piravom,Ernakulam

Age: 26

Employee ID: 45

Company Name: Infosys

Qualification: BCom

Salary: 55000

Subject: DBMS

Department: CS

Teacher ID: 56

Details of all teachers:

Teacher 1:

Name: Akhila

Gender: Female

Address: ABC(H),Keezhoor,Kottayam

Age: 24

Employee ID: 2244

Company Name: TCS

Qualification: MCA

Salary: 45000.0

Subject: Maths

Department: BCA

Teacher ID: 34

Teacher 2:

Name: Anu

Gender: Female

Address: MNO(H).

,Piravom,Ernakulam

Age: 26

Employee ID: 45

Company Name: Infosys

Qualification: BCom

Salary: 55000.0

Subject: DBMS

Department: CS

Teacher ID: 56

Experiment 17**Date: 19.04.2024****Interface - 1****Aim:**

Create an interface having prototypes of functions area() and perimeter(). Create two classes Circle and Rectangle which implements the above interface. Create a menu driven program to find area and perimeter of objects.

Program

```
import java.util.Scanner;

interface Shape {

    double area();

    double perimeter();

}

class Circle implements Shape {

    private double radius;

    public Circle(double radius) {

        this.radius = radius;

    }

    public double area() {

        return Math.PI * radius * radius;

    }

    public double perimeter() {

        return 2 * Math.PI * radius;

    }

}

class Rectangle implements Shape {

    private double length;

    private double width;

    public Rectangle(double length, double width) {

        this.length = length;

        this.width = width;

    }

}
```

```
public double area() {
    return length * width;
}

public double perimeter() {
    return 2 * (length + width);
}
}

public class ShapeCalculator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        while (true) {
            System.out.println("1. Calculate area and perimeter of a circle");
            System.out.println("2. Calculate area and perimeter of a rectangle");
            System.out.println("3. Exit");
            System.out.print("Enter your choice: ");
            int choice = scanner.nextInt();
            if (choice == 1) {
                System.out.print("Enter the radius of the circle: ");
                double radius = scanner.nextDouble();
                Circle circle = new Circle(radius);
                System.out.println("Area of the circle: " + circle.area());
                System.out.println("Perimeter of the circle: " + circle.perimeter());
            } else if (choice == 2) {
                System.out.print("Enter the length of the rectangle: ");
                double length = scanner.nextDouble();
                System.out.print("Enter the width of the rectangle: ");
                double width = scanner.nextDouble();
                Rectangle rectangle = new Rectangle(length, width);
                System.out.println("Area of the rectangle: " + rectangle.area());
                System.out.println("Perimeter of the rectangle: " + rectangle.perimeter());
            } else if (choice == 3) {
```

```
break;
} else {
System.out.println("Invalid choice. Please try again.");
}
}
scanner.close();
}
}
```

Output

1. Calculate area and perimeter of a circle
2. Calculate area and perimeter of a rectangle
3. Exit

Enter your choice: 1

Enter the radius of the circle: 2

Area of the circle: 12.5663

Perimeter of the circle: 12.5663

1. Calculate area and perimeter of a circle
2. Calculate area and perimeter of a rectangle
3. Exit

Enter your choice: 2

Enter the length of the rectangle: 3

Enter the width of the rectangle: 5

Area of the rectangle: 15.0

Perimeter of the rectangle: 16.0

1. Calculate area and perimeter of a circle
2. Calculate area and perimeter of a rectangle
3. Exit

Enter your choice: 3

Experiment 18**Date: 19.04.2024****Interface - 2****Aim:**

Prepare bill with the given format using calculate method from interface. Order No.

Date:

Product Id	Name	Quantity	unit price	Total
------------	------	----------	------------	-------

101	A	2	25.0	50.0
-----	---	---	------	------

102	B	1	100.0	100.0
-----	---	---	-------	-------

Net.Amount 150.0

Program

```
import java.util.Scanner;

interface Calc {
    void calculate();
}

class Bill implements Calc {
    String date, name, p_id;
    int quantity;
    double unit_price, total, namount = 0;
    Scanner sc = new Scanner(System.in);

    public void getdata() {
        System.out.println("Enter product id:");
        p_id = sc.nextLine();
        System.out.println("Enter product name:");
        name = sc.nextLine();
        System.out.println("Enter the Quantity:");

        quantity = sc.nextInt();
        System.out.println("Enter the unit price:");
```

```
unit_price = sc.nextDouble();
}
public void calculate() {
total = quantity * unit_price;
}
public void display() {
System.out.println(p_id + "\t\t" + name + "\t\t" + quantity + "\t\t" + unit_price + "\t" +
total);
}
}
public class BillingSystem {
public static void main(String args[]) {
int n, i;
double namount = 0, t;
String date;
String orno;
Scanner sc = new Scanner(System.in);
System.out.println("Enter the Order no:");
orno = sc.nextLine();
System.out.println("Enter the date:");
date = sc.nextLine();
System.out.println("Enter how many products are there:");
n = sc.nextInt();
Bill ob[] = new Bill[n];
for (i = 0; i < n; i++)
ob[i] = new Bill();
for (i = 0; i < n; i++) {
ob[i].getdata();
ob[i].calculate();
}
System.out.println("\n-----Bill Details-----");
```

```
System.out.println("Date:" + date);
System.out.println("Order No:" + orno);
System.out.println("Product Id \tName\t Quantity\t unit price\t Total ");
System.out.println("-----");
for (i = 0; i < n; i++) {
    ob[i].display();
    namount += ob[i].total;
}
System.out.println("-----");
System.out.println("\t\t\tNet.Amount\t" + namount);
}
}
```

Output

Enter the Order no:

#3902

Enter the date:

12/05/24

Enter how many products are there:

2

Enter product id:

101

Enter product name:

A

Enter the Quantity:

2

Enter the unit price:

25

Enter product id:

102

Enter product name:

B

Enter the Quantity:

1

Enter the unit price:

100

-----Bill Details-----

Date:12/05/24

Order No:#3902

Product Id	Name	Quantity	unit price	Total
------------	------	----------	------------	-------

101	A	2	25.0	50.0
-----	---	---	------	------

102	B	1	100.0	100.0
-----	---	---	-------	-------

Net.Amount				150.0
------------	--	--	--	-------

Experiment 19**Date: 24.04.2024****Packages - 1****Aim:**

Create a Graphics package that has classes for shapes Rectangle, Triangle, Square and Circle. Test the package by finding the area of these figures.

Hint:- Create 3 java files for calculate the area 3 different shapes in the directory Shapes inside the directory where the java program is stored. Then import all the class files inside the package Shapes to our original program.

Equation for area of a circle= $A=\pi r^2$.

Area of a triangle = $\sqrt{s(s-a)(s-b)(s-c)}$

Area of a rectangle= $l*b$

Program

```
// Graphics/Rectangle.java
```

```
package Graphics;

public class Rectangle {
    private double length;
    private double breadth;

    public Rectangle(double length, double breadth) {
        this.length = length;
        this.breadth = breadth;
    }

    public double calculateArea() {
        return length * breadth;
    }
}
```

```
// Graphics/Triangle.java
```

```
package Graphics;

public class Triangle {
    private double sideA;
    private double sideB;
    private double sideC;

    public Triangle(double sideA, double sideB, double sideC) {
        this.sideA = sideA;
    }
}
```

```
this.sideB = sideB;
this.sideC = sideC;
}
public double calculateArea() {
double s = (sideA + sideB + sideC) / 2;
return Math.sqrt(s * (s - sideA) * (s - sideB) * (s - sideC));
}
}
```

```
// Graphics/Square.java
```

```
package Graphics;
public class Square {
private double side;
public Square(double side) {
this.side = side;
}
public double calculateArea() {
return side * side;
}
}
```

```
// Graphics/Circle.java
```

```
package Graphics;
public class Circle {
private double radius;
public Circle(double radius) {
this.radius = radius;
}
public double calculateArea() {
return Math.PI * radius * radius;
}
}
```

```
// MainGraphics.java
```

```
import java.util.Scanner;
import Graphics.*;
public class MainGraphics {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter dimensions for Rectangle (length breadth):");
        double lengthRectangle = scanner.nextDouble();
        double breadthRectangle = scanner.nextDouble();
        Rectangle rectangle = new Rectangle(lengthRectangle, breadthRectangle);
        System.out.println("Area of Rectangle: " + rectangle.calculateArea());
        System.out.println("Enter dimensions for Triangle (sideA sideB sideC):");
        double sideA = scanner.nextDouble();
        double sideB = scanner.nextDouble();
        double sideC = scanner.nextDouble();
        Triangle triangle = new Triangle(sideA, sideB, sideC);
        System.out.println("Area of Triangle: " + triangle.calculateArea());
        System.out.println("Enter dimension for Square (side):");
        double sideSquare = scanner.nextDouble();
        Square square = new Square(sideSquare);
        System.out.println("Area of Square: " + square.calculateArea());
        System.out.println("Enter radius for Circle:");
        double radiusCircle = scanner.nextDouble();
        Circle circle = new Circle(radiusCircle);
        System.out.println("Area of Circle: " + circle.calculateArea());
        scanner.close();
    }
}
```

Output

Enter dimensions for Rectangle (length breadth):

5 4

Area of Rectangle: 20.0

Enter dimensions for Triangle (sideA sideB sideC):

3 4 5

Area of Triangle: 6.0

Enter dimension for Square (side):

6

Area of Square: 36.0

Enter radius for Circle:

7

Area of Circle: 153.9

Experiment 20**Date: 24.04.2024****Packages - 2****Aim:**

Create an Arithmetic package that has classes for the 4 basic arithmetic operations. Test the package by implementing all operations on two given numbers.

Program

```
// Arithmetic/Addition.java
```

```
package Arithmetic;

public class Addition {

    public static double add(double num1, double num2) {

        return num1 + num2;

    }

}
```

```
// Arithmetic/Subtraction.java
```

```
package Arithmetic;

public class Subtraction {

    public static double subtract(double num1, double num2) {

        return num1 - num2;

    }

}
```

```
// Arithmetic/Multiplication.java
```

```
package Arithmetic;

public class Multiplication {

    public static double multiply(double num1, double num2) {

        return num1 * num2;

    }

}
```

```
// Arithmetic/Division.java
```

```
package Arithmetic;

public class Division {

    public static double divide(double num1, double num2) {
```

```
if (num2 == 0) {  
    System.out.println("Error! Division by zero is not allowed.");  
    return 0;  
}  
return num1 / num2;  
}  
}  
  
// MainProgram.java  
import java.util.Scanner;  
import Arithmetic.*;  
public class MainProgram {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        System.out.print("Enter first number: ");  
        double num1 = scanner.nextDouble();  
        System.out.print("Enter second number: ");  
        double num2 = scanner.nextDouble();  
        System.out.println("Addition: " + Addition.add(num1, num2));  
        System.out.println("Subtraction: " + Subtraction.subtract(num1, num2));  
        System.out.println("Multiplication: " + Multiplication.multiply(num1, num2));  
        System.out.println("Division: " + Division.divide(num1, num2));  
        scanner.close();  
    }  
}
```

Output

Enter first number: 10

Enter second number: 5

Addition: 15.0

Subtraction: 5.0

Multiplication: 50.0

Division: 2.0

Experiment 21**Date: 24.04.2024****Exception Handling - 1****Aim:**

Write a user defined exception class to authenticate the user name and password.

Program

```
import java.util.Scanner;

class AuthenticationException extends Exception {
    public AuthenticationException(String message) {
        super(message);
    }
}

class Authentication {
    private static final String USERNAME = "user123";
    private static final String PASSWORD = "password156";

    public static void authenticate(String username, String password) throws
        AuthenticationException {
        if (!USERNAME.equals(username) || !PASSWORD.equals(password)) {
            throw new AuthenticationException("Invalid username or password!");
        }

        System.out.println("Authentication successful");
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter username: ");
        String username = scanner.nextLine();

        System.out.print("Enter password: ");

        String password = scanner.nextLine();
        try {
```

```
Authentication.authenticate(username, password);  
} catch (AuthenticationException e) {  
System.out.println("Authentication failed: " + e.getMessage());  
}  
scanner.close();  
}  
}
```

Output

Enter username: user

Enter password: pass

Authentication failed: Invalid username or password!

Experiment 22**Date: 24.04.2024****Exception Handling - 2****Aim:**

Find the average of N positive integers, raising a user defined exception for each negative input.

Program

```
import java.util.Scanner;

class NegativeInputException extends Exception {
    public NegativeInputException(String message) {
        super(message);
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of integers: ");
        int n = scanner.nextInt();
        int sum = 0;
        int count = 0;
        for (int i = 1; i <= n; i++) {
            try {
                System.out.print("Enter integer " + i + ": ");
                int num = scanner.nextInt();
                if (num < 0) {
                    throw new NegativeInputException("Negative number entered!");
                }
                sum += num;
            } catch (NegativeInputException e) {
                System.out.println(e);
            }
        }
        count++;
    }
}
```

```
i--;  
}  
}  
if (count > 0) {  
    double average = (double) sum / count;  
    System.out.println("Average of positive integers: " + average);  
} else {  
    System.out.println("No positive integers entered!");  
}  
scanner.close();  
}  
}
```

Output

```
Enter the number of integers: 3  
Enter integer 1: -8  
NegativeInputException: Negative number entered!  
Enter integer 1: 5  
Enter integer 2: 7  
Enter integer 3: 8  
Average of positive integers: 6.666666666666667
```

Experiment 23**Date: 24.04.2024****Exception Handling - 3****Aim:**

Program to find the sum of command line arguments and count the invalid integers entered through command line.

Program

```
import java.util.Scanner;

public class UserInputSum {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int sum = 0;
        int invalidCount = 0;
        System.out.println("Enter integers separated by spaces and press Enter:");
        String inputLine = scanner.nextLine();
        String[] inputNumbers = inputLine.split("\\s+");
        for (String input : inputNumbers) {
            try {
                int num = Integer.parseInt(input);
                sum += num;
            } catch (NumberFormatException e) {
                System.out.println(e);
                invalidCount++;
            }
        }
        System.out.println("Sum of valid integers: " + sum);
        System.out.println("Count of invalid integers: " + invalidCount);
        scanner.close();} }
```

Output

Enter integers separated by spaces and press Enter:

3 a 5 6

java.lang.NumberFormatException: For input string: "a"

Sum of valid integers: 14

Count of invalid integers: 1

Course Outcome 4**Experiment 24****Date: 04.05.2024****File Handling - 1****Aim:**

Program to list the sub directories and files in a given directory and also search for a file name.

Program

```
import java.io.File;
```

```
public class FileSubDir {  
    static void RecursivePrint(File[] arr, int index, int level) {  
        if (index == arr.length)  
            return;  
        for (int i = 0; i < level; i++)  
            System.out.print("\t");  
        if (arr[index].isFile())  
            System.out.println(arr[index].getName());  
        else if (arr[index].isDirectory()) {  
            System.out.println "[" + arr[index].getName() + " ]";  
            RecursivePrint(arr[index].listFiles(), 0, level + 1);  
        }  
        RecursivePrint(arr, ++index, level);  
    }  
    static void searchFile(File[] arr, String fileName) {  
        if (arr == null || arr.length == 0)  
            return;  
        int length = arr.length;  
        for (int i = 0; i < length; i++) {  
            File file = arr[i];  
            if (file.isDirectory()) {
```

```
        searchFile(file.listFiles(), fileName);
    } else {
        if (file.getName().equals(fileName)) {
            System.out.println("File found: " + file.getAbsolutePath());
        }
    }
}
}

public static void main(String[] args) {
    String maindirpath = "C:/Users/Meenus/Downloads/Meenus";
    String searchFileName = "provisional.PDF";
    File maindir = new File(maindirpath);
    if (maindir.exists() && maindir.isDirectory()) {
        File arr[] = maindir.listFiles();
        System.out.println("-----");
        System.out.println("Files from main directory : " + maindir);
        System.out.println("-----");
        RecursivePrint(arr, 0, 0);
        System.out.println("\nSearching for file: " + searchFileName);
        searchFile(arr, searchFileName);
    }
}
}
```

Output

PS C:\Users\Meenus\Desktop> javac FileSubDir.java

PS C:\Users\Meenus\Desktop> java FileSubDir

Files from main directory : C:\Users\Meenus\Downloads\Meenus

1657947568110.jpg

Akhila aadhaar.pdf

akhila degree crtftct.PDF

Akhila electionid.pdf

Akhila offical.jpg

Birth Certificate.pdf

Income certificate.pdf

Nativity.pdf

NCL.pdf

plustwo.PDF

provisional.PDF

sslc.PDF

voter ID.pdf

Searching for file: provisional.PDF

File found: C:\Users\Meenus\Downloads\Meenus\provisional.PDF

Experiment 25**Date: 04.05.2024****File Handling - 2****Aim:**

Write a program to write to a file, then read from that file and display the contents on the console.

Program

```
import java.io.*;

public class FileReadWriteExample {
    public static void main(String args[]) {
        try {
            FileWriter writer = new FileWriter("sample.txt");
            writer.write("Hello, World!");
            writer.close();
            System.out.println("Write successful!");
        } catch (Exception e) {
            System.out.println("An error occurred while writing to the file.");
            e.printStackTrace();
        }

        try {
            FileReader reader = new FileReader("sample.txt");
            BufferedReader bufferedReader = new BufferedReader(reader);
            String line;
            System.out.println("Contents of the file:");
            while ((line = bufferedReader.readLine()) != null) {
                System.out.println(line);
            }
            bufferedReader.close();
        } catch (Exception e) {
            System.out.println("An error occurred while reading the file.");
            System.out.println(e);
        }
    }
}
```

```
}  
}
```

Output

```
C:\Users\Meenus\Downloads> javac FileReadWriteExample.java
```

```
C:\Users\Meenus\Downloads> java FileReadWriteExample
```

Write successful!

Contents of the file:

Hello, World!

Experiment 26**Date: 04.05.2024****File Handling - 3****Aim:**

Write a program to copy one file to another.

Program

```
import java.io.*;

public class FileCopy {
    public static void main(String args[]) {
        String sourceFilePath = "source.txt";
        String destinationFilePath = "destination.txt";
        try {
            FileInputStream inputStream = new FileInputStream(sourceFilePath);
            FileOutputStream outputStream = new FileOutputStream(destinationFilePath);
            byte[] buffer = new byte[1024];
            int length;
            while ((length = inputStream.read(buffer)) > 0) {
                outputStream.write(buffer, 0, length);
            }
            inputStream.close();
            outputStream.close();
            System.out.println("File copied successfully.");
        } catch (Exception e) {
            System.out.println("An error occurred while copying the file.");
            System.out.println(e);
        }
    }
}
```

Output

```
C:\Users\Meenus\Downloads> javac FileCopy.java
```

```
C:\Users\Meenus\Downloads> java FileCopy
```

File copied successfully.

Experiment 27**Date: 04.05.2024****File Handling - 4****Aim:**

Write a program that reads from a file having integers. Copy even numbers and odd numbers to separate files.

Program

```
import java.io.*;
public class SeparateNum {
    public static void main(String[] args) {
        String inputFilePath = "numbers.txt";
        String evenOutputFilePath = "even_numbers.txt";
        String oddOutputFilePath = "odd_numbers.txt";
        try {
            BufferedReader reader = new BufferedReader(new FileReader(inputFilePath));
            BufferedWriter evenWriter = new BufferedWriter(new FileWriter(evenOutputFilePath));
            BufferedWriter oddWriter = new BufferedWriter(new FileWriter(oddOutputFilePath));
            String line;
            while ((line = reader.readLine()) != null) {
                int number = Integer.parseInt(line);
                if (number % 2 == 0) {
                    evenWriter.write(Integer.toString(number));
                    evenWriter.newLine();
                } else {
                    oddWriter.write(Integer.toString(number));
                    oddWriter.newLine();
                }
            }
            reader.close();
            evenWriter.close();

            oddWriter.close();
            System.out.println("Even and odd numbers separated successfully.");
        } catch (Exception e) {
            System.out.println("An error occurred while processing the file.");
            System.out.println(e);
        }
    }
}
```

Output

```
C:\Users\Meenus\Downloads> javac SeparateNum.java
```

```
C:\Users\Meenus\Downloads> java SeparateNum
```

Even and odd numbers separated successfully.

Experiment 28**Date: 04.05.2024****File Handling - 5****Aim:**

Write a program to create a generic stack and do the Push and Pop operations.

Program

```
import java.util.*;

class operations{

public void operation()

{

int top =-1,ch,n,e;

Scanner inp = new Scanner(System.in);

System.out.println("Enter Size of Stack :");

n = inp.nextInt();

int size=n-1;

int[] arr = new int[n];

do {

System.out.println("-----\nMENU : \n1.push \n2.pop \n3.Display \n4.Exit \n-----");

System.out.println("Enter your choice");

ch = inp.nextInt();

switch(ch)

{

case 1 :

if(top == size)

{

System.out.println("Stack is Full");

}

else

{

System.out.println("Enter Element : ");
```

```
e = inp.nextInt();
top++;
arr[top] = e;
}
break;
case 2 :
if(top == -1)
{
System.out.println("Stack is empty");
}
else
{
System.out.println(arr[top] + " is removed ");
top--;
}
break;
case 3 :
if(top == -1)
{
System.out.println("Stack is empty");
}
else
{
System.out.println("Stack: ");
for(int i=top; i>=0; i--)
{

System.out.println(arr[i]);
}
}
break;
```


case 4 :

```
System.exit(0);
default : System.out.println("Invalid Choice");
}
}while(ch !=4);
}
}
public class MainStack {
public static void main(String[] args) {
operations obj = new operations();
obj.operation();
}
}
```

Output

Enter Size of Stack :

5

MENU :

1.push

2.pop

3.Display

4.Exit

Enter your choice

1

Enter Element :

3

MENU :

1.push

2.pop

3.Display

4.Exit

Enter your choice

1

Enter Element :

7

MENU :

1.push

2.pop

3.Display

4.Exit

Enter your choice

1

Enter Element :

9

MENU :

1.push

2.pop

3.Display

4.Exit

Enter your choice

3

Stack:

9

7

3

MENU :

1.push

2.pop

3.Display

4.Exit

Enter your choice

2

9 is removed

MENU :

1.push

2.pop

3.Display

4.Exit

Enter your choice

3

Stack:

7

3

MENU :

1.push

2.pop

3.Display

4.Exit

Enter your choice

4

Experiment 29**Date: 08.05.2024****File Handling - 6****Aim:**

Using generic method perform Bubble sort.

Program

```
import java.util.*;

class Bubblesort {
void sort(int arr[])
{
int n = arr.length;
for(int i =0;i < n-1;i++)
{
for(int j=0;j<n-i-1;j++)
{
if(arr[j] > arr[j+1])
{
int temp = arr[j];
arr[j]=arr[j+1];
arr[j+1]= temp;
}
}
}
}

void display(int arr[])
{
System.out.println("Sorted Array :");
int n = arr.length;

for(int i=0;i<n;i++)
{
```

```
System.out.print(arr[i]+ " ");  
}  
}  
public static void main(String[] args)  
{  
    int n,e;  
    System.out.println("Enter size of Array :");  
    Scanner inp =new Scanner(System.in);  
    n = inp.nextInt();  
    int[] arr = new int[n];  
    for(int i=0;i<n;i++)  
    {  
        System.out.println("Enter element :");  
        e = inp.nextInt();  
        arr[i]=e;  
    }  
    Bubblesort ob = new Bubblesort();  
    ob.sort(arr);  
    ob.display(arr);  
}  
}
```

Output

Enter size of Array :

7

Enter element :

3

Enter element :

5

Enter element :

6

Enter element :

1

Enter element :

2

Enter element :

9

Enter element :

5

Sorted Array :

1 2 3 5 5 6 9

Experiment 30**Date: 08.05.2024****File Handling - 7****Aim:**

Maintain a list of Strings using ArrayList from collection framework, perform built-in operations.

Program

```
import java.util.*;

public class ArrayListex {

    public static void main(String args[]) {
        ArrayList<String> obj = new ArrayList<String>();
        obj.add("Akhila");
        obj.add("Meenu");
        obj.add("Rahul");
        obj.add("Arjun");
        System.out.println("Original ArrayList:");
        for(String str:obj)
            System.out.println(str);
        obj.add(1, "Anu");
        System.out.println("ArrayList after add operation:");
        for(String str:obj)
            System.out.println(str);
        obj.remove("Anu");
        System.out.println("ArrayList after remove operation:");
        for(String str:obj)
            System.out.println(str);
        obj.remove(3);
        System.out.println("Final ArrayList:");

        for(String str:obj)
            System.out.println(str);
    }
}
```

```
Collections.sort(obj);  
System.out.println("ArrayList after sorting:");  
for (String str : obj)  
System.out.println(str);  
System.out.println("Object at index 2:"+obj.get(2));  
System.out.println("Six is in the ArrayList :"+obj.contains("Binu"));  
System.out.println("Two is in the ArrayList :"+obj.contains("Meenu"));  
System.out.println("Size of the ArrayList:"+obj.size());  
obj.clear();  
System.out.println("ArrayList Removed");  
}  
}
```

Output

Original ArrayList:

Akhila

Meenu

Rahul

Arjun

ArrayList after add operation:

Akhila

Anu

Meenu

Rahul

Arjun

ArrayList after remove operation:

Akhila

Meenu

Rahul

Arjun

Final ArrayList:

Akhila

Meenu

Rahul

ArrayList after sorting:

Akhila

Meenu

Rahul

Object at index 2:Rahul

Six is in the ArrayList :false

Two is in the ArrayList :true

Size of the ArrayList:3

ArrayList Removed

Experiment 31**Date: 08.05.2024****File Handling - 8****Aim:**

Program to remove all the elements from a linked list

Program

```
import java.util.*;

public class Linkedlistex {

    public static void main(String[] args){

        LinkedList<String> L=new LinkedList<>();

        L.add("Red");

        L.add("Green");

        L.add("Blue");

        L.add(0,"Colors");

        System.out.println(L);

        L.remove("Blue");

        System.out.println(L);

        L.remove(2);

        System.out.println(L);

        L.removeLast();

        System.out.println(L);

        L.removeFirst();

        System.out.println(L);

    }

}
```

Output

[Colors, Red, Green, Blue]

[Colors, Red, Green]

[Colors, Red]

[Colors]

[]

Experiment 32**Date: 08.05.2024****File Handling - 9****Aim:**

Program to remove an object from the Stack when the position is passed as parameter

Program

```
import java.util.*;

public class Stackremove {

    public static void main(String[] args) {

        Stack<Integer> st = new Stack<>();

        st.push(101);

        st.push(202);

        st.push(303);

        st.push(404);

        System.out.println("Stack: "+st);

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the position : ");

        int x = sc.nextInt();

        st.remove(x);

        System.out.println("Stack: "+st);

    }

}
```

Output

Stack: [101, 202, 303, 404]

Enter the position :

2

Stack: [101, 202, 404]

Experiment 33**Date: 08.05.2024****Multithreading - 1****Aim:**

Define 2 classes; one for generating multiplication table of 5 and other for displaying first N prime numbers. Implement using threads. (Thread class)

Program

```
import java.util.Scanner;

class mul extends Thread{

public void run(){

for(int i =0;i<11;i++){

System.out.println("5*" +i+" = "+5*i);

}

}

}

class prime extends Thread{

int num;

public prime(int n){

this.num=n;

}

public void run(){

int x, y, flg;

System.out.println("All the Prime numbers within 1 and " + num + " are:");

for (x = 1; x <= num; x++) {

if (x == 1 || x == 0)

continue;

flg = 1;

for (y = 2; y <= x / 2; ++y) {

if (x % y == 0) {

flg = 0;

break;

}
```

```
}  
}  
if (flg == 1)  
System.out.print(x + " ");  
}  
System.out.println("\n");  
}  
}  
public class MainThread {  
    public static void main(String args[]) {  
        System.out.println("Enter The number:");  
        Scanner sc = new Scanner(System.in);  
        int number = sc.nextInt();  
        mul obj1 = new mul();  
        obj1.start();  
        prime obj2 = new prime(number);  
        obj2.start();  
    }  
}
```

Output

Enter The number:

7

All the Prime numbers within 1 and 7 are:

2 3 5 7

5*0 = 0

5*1 = 5

5*2 = 10

5*3 = 15

5*4 = 20

$$5*5 = 25$$

$$5*6 = 30$$

$$5*7 = 35$$

$$5*8 = 40$$

$$5*9 = 45$$

$$5*10 = 50$$

Experiment 34**Date: 08.05.2024****Multithreading - 2****Aim:**

Define 2 classes; one for generating Fibonacci numbers and other for displaying even numbers in a given range. Implement using threads. (Runnable Interface)

Program

```
import java.util.Scanner;
class MainNum {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the lower limit of range");
        int x = sc.nextInt();
        System.out.println("Enter the Upper limit of range");
        int y = sc.nextInt();
        Runnable r = new fibannoci(x,y);
        Thread obj1 = new Thread(r);
        obj1.start();
        Runnable p = new even(x,y);
        Thread obj2 = new Thread(p);
        obj2.start();
    }
}
class fibannoci implements Runnable{
    int n1;
    int n2;
    int num =0;
    int x = 0;
    int y = 1;
    fibannoci(int l,int u){

        n1 = l;
        n2 = u;
    }
    @Override
    public void run() {
        System.out.println("fibannoci = "+0);
        System.out.println("fibannoci = "+1);
        while (num < n2-1) {
            num = x + y;
            if ((num >= n1)&&(num<=n2)) {
                System.out.println("fibannoci = "+num);
            }
            x = y;
```



```
y = num;}}  
}  
class even implements Runnable{  
int n1;  
int n2;  
even(int x, int y){  
n1 = x;  
n2 = y;  
}  
@Override  
public void run(){  
for(int i= n1; i<=n2; i++){  
if(i%2==0){  
System.out.println("even =" +i);  
}  
}}  
}
```

Output

Enter the lower limit of range

20

Enter the Upper limit of range

30

fibannoci = 0

fibannoci = 1

fibannoci = 21

even =20

even =22

even =24

even =26

even =28

even =30

