**CLASS AND OBJECT**

**File name:Student.java**

class Student {

String name;

int rno;

double percentage;

void setDetails(String studname, int roll, double perc) {

name = studname;

rno = roll;

percentage = perc;

}

void displayDetails() {

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

System.out.println("Roll Number: " + rno);

System.out.println("Marks: " + percentage);

}

public static void main(String[] args) {

Student s1 = new Student();

Student s2 = new Student();

s1.setDetails("Mephisto", 101, 88.5);

s2.setDetails("Peter", 102, 92.0);

System.out.println("Student 1:");

s1.displayDetails();

System.out.println("\nStudent 2:");

s2.displayDetails(); } }

**USING CONSTRUCTOR**

**Filename:Employee.java**

class emp {

int id;

String name;

String department;

double salary;

emp()

{

id=101;

name="Mephisto";

department="CYS";

salary=10000;

}

emp(int id, String name, String department, double salary) {

this.id = id;

this.name = name;

this.department = department;

this.salary = salary;

}

void displayInfo() {

System.out.println("Employee ID: " + id);

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

System.out.println("Department: " + department);

System.out.println("Salary : $" + salary);

System.out.println("-----------------------------------");

}

}

class Employee {

public static void main(String[] args) {

emp e1=new emp();

emp emp1 = new emp(101, "Lucifer", "HR", 55000.00);

emp emp2 = new emp(102, "Beliah", "IT", 72000.00);

emp emp3 = new emp(103, "Tony Stark", "Finance", 68000.00);

e1.displayInfo();

emp1.displayInfo();

emp2.displayInfo();

emp3.displayInfo();

}

}

**USING COMMAND - LINE ARGUMENTS**

**Filename:sum.java**

class sum {

public static void main(String[] args) {

try

{

// Parse two numbers from command-line arguments

int num1 = Integer.parseInt(args[0]);

int num2 = Integer.parseInt(args[1]);

// Calculate and print the sum

int sum = num1 + num2;

System.out.println("Sum = " + sum);

}

catch (Exception e)

{

System.out.println("Error: Please provide two valid integers as command-line arguments.");

System.out.println("Usage: java Sum <number1> <number2>");

}

}

}

**USING VECTOR**

**Filename:VectorExample.java**

import java.util.Vector;

class VectorExample {

public static void main(String[] args) {

Vector<String> fruits = new Vector<>();

fruits.add("Apple");

fruits.add("Banana");

fruits.add("Mango");

fruits.add("Orange");

System.out.println("Fruits in the vector:");

for (String ff : fruits) {

System.out.println(ff);

}

fruits.remove("Banana");

System.out.println("\nAfter removing 'Banana':");

for (String ff : fruits) {

System.out.println(ff);

}

System.out.println("\nFruit at index 1: " + fruits.get(1));

System.out.println("Total number of fruits: " + fruits.size());

}

}

**USING INTERFACE**

**Filename:Shape.java**

public interface Shape {

double calculateArea();

double calculatePerimeter();

}

**Filename:Circle.java**

public class Circle implements Shape

{

private double radius;

public Circle(double radius) {

this.radius = radius;

}

public double calculateArea() {

return Math.PI \* radius \* radius;

}

public double calculatePerimeter() {

return 2 \* Math.PI \* radius;

}

}

**Filename:Rectangle.java**

public class Rectangle implements Shape {

private double length;

private double width;

public Rectangle(double length, double width) {

this.length = length;

this.width = width;

}

public double calculateArea() {

return length \* width;

}

public double calculatePerimeter() {

return 2 \* (length + width);

}

}

**RUN THIS🡺Filename:ShapeDemo.java**

public class ShapeDemo {

public static void main(String[ ] args) {

Shape circle = new Circle(7.0);

Shape rectangle = new Rectangle(5.0, 3.0);

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

System.out.println("Circle Perimeter: " + circle.calculatePerimeter());

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

System.out.println("Rectangle Perimeter: " + rectangle.calculatePerimeter());

}

}

**USING ALL FORMS OF INHERITANCE**

**Filename:Inheritance.java**

class Person {

String name;

int age;

void setPerson(String n, int a) {

name = n;

age = a;

}

void displayPerson() {

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

System.out.println("Age: " + age);

}

}

class Student extends Person {

int m1, m2, m3;

void setMarks(int a, int b, int c) {

m1 = a;

m2 = b;

m3 = c;

}

void calculateAverage() {

float avg = (m1 + m2 + m3) / 3.0f;

System.out.println("Average Marks: " + avg);

}

}

class Graduate extends Student {

void checkPass() {

if (m1 >= 40 && m2 >= 40 && m3 >= 40)

System.out.println("Result: Pass");

else

System.out.println("Result: Fail");

}

}

class Staff extends Person {

double basicSalary;

void setSalary(double salary) {

basicSalary = salary;

}

void calculateSalary() {

double hra = 0.10 \* basicSalary;

double da = 0.05 \* basicSalary;

double total = basicSalary + hra + da;

System.out.println("Total Salary: Rs. " + total);

}

}

class Inheritance {

public static void main(String args[]) {

System.out.println("----- Graduate Student -----");

Graduate g = new Graduate();

g.setPerson("Alice", 20);

g.setMarks(75, 82, 67);

g.displayPerson();

g.calculateAverage();

g.checkPass();}

}

**USING STRING AND STRINGBUFFER CLASS**

**Filename:StringDemo.java**

import java.util.Scanner;

public class StringDemo

{

public static void main(String[] args)

{

Scanner scanner = new Scanner(System.in);

System.out.println("=== STRING CLASS OPERATIONS ===");

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

String name = scanner.nextLine();

System.out.println("Length of name: " + name.length());

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

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

System.out.println("Does name contain 'a'? " + name.contains("a"));

System.out.println("Character at position 2: " + name.charAt(2));

String fullName = name.concat(" Kumar");

System.out.println("After concatenation: " + fullName);

System.out.println("Starts with 'La'? " + name.startsWith("La"));

System.out.println("Ends with 'ya'? " + name.endsWith("ya"));

System.out.println("Index of 'a': " + name.indexOf('a'));

System.out.println("Last index of 'a': " + name.lastIndexOf('a'));

System.out.println("Trimmed string: '" + (" " + name + " ").trim() + "'");

System.out.println("String equals 'Lavanya'? " + name.equals("Lavanya"));

System.out.println("String equalsIgnoreCase 'lavanya'? " + name.equalsIgnoreCase("lavanya"));

System.out.println("\n=== STRINGBUFFER CLASS OPERATIONS ===");

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

String message = scanner.nextLine();

StringBuffer buffer = new StringBuffer(message);

buffer.append(" - Welcome!");

System.out.println("After appending: " + buffer);

buffer.insert(0, "Hello, ");

System.out.println("After insertion: " + buffer);

buffer.replace(0, 5, "Hi");

System.out.println("After replacement: " + buffer);

buffer.reverse();

System.out.println("After reversing: " + buffer);

buffer.reverse();

buffer.delete(0, 3);

System.out.println("After deleting first 3 characters: " + buffer);

System.out.println("Capacity of buffer: " + buffer.capacity());

buffer.ensureCapacity(100);

System.out.println("New capacity after ensureCapacity(100): " + buffer.capacity());

System.out.println("Length of buffer: " + buffer.length());

scanner.close();

}

}

**EXCEPTION HANDLING**

**Filename:Error.java**

import java.util.\*;

class Error{

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

try {

System.out.print("Enter your roll number (integer): ");

int rollNo = scanner.nextInt();

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

int m = scanner.nextInt();

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

int n = scanner.nextInt();

int div = m/n;

System.out.println("Division =: " + div);

int[] scores = {85, 90, 78};

System.out.print("Enter subject index to view marks : ");

int index = scanner.nextInt();

System.out.println("Subject " + index + " marks: " + scores[index]);

String name = null;

System.out.println("Student name length: " + name.length());

scanner.nextLine();

System.out.print("Enter age (as string): ");

String ageInput = scanner.nextLine();

int age = Integer.parseInt(ageInput);

System.out.println("Student age: " + age);

String grade = "A+";

System.out.println("Grade character at index 5: " + grade.charAt(5));

System.out.print("Enter size of array: ");

int size = scanner.nextInt();

int[] dynamicArray = new int[size];

System.out.println("Array of size " + size + " created.");

}

catch (InputMismatchException e) {

System.out.println("Error: Invalid input type. Please enter correct data.");

}

catch (ArithmeticException e) {

System.out.println("Error: Division by zero is not allowed.");

}

catch (ArrayIndexOutOfBoundsException e) {

System.out.println("Error: You tried to access an invalid index of the array.");

}

catch (NullPointerException e) {

System.out.println("Error: Student name is not initialized (null reference).");

}

catch (NumberFormatException e) {

System.out.println("Error: Cannot convert the entered string into a number.");

}

catch (StringIndexOutOfBoundsException e) {

System.out.println("Error: Invalid index used to access a character in string.");

}

catch (NegativeArraySizeException e) {

System.out.println("Error: Array size cannot be negative.");

}

finally {

System.out.println("Finally block. Thank you!");

scanner.close();

}

}

}

**IMPLEMENTING THREAD BASED APPLICATIONS**

**Filename:** **ThreadExample.java**

class MyThread extends Thread {

public void run() {

for (int i = 1; i <= 5; i++) {

System.out.println("Child Thread: " + i);

try {

Thread.sleep(500);

} catch (InterruptedException e) {

System.out.println(e);

}

}

}

}

class ThreadExample1 {

public static void main(String[] args) {

MyThread t1 = new MyThread();

t1.start( );

for (char c='A'; c<='F'; c++) {

System.out.println("Main Thread: " + c);

try {

Thread.sleep(1000);

}

catch (InterruptedException e) {

System.out.println(e);

}

}

}

}

**PAKAGES**

**Filename: Circle.java**

package shapes;

public class Circle {

private double radius;

public Circle(double radius) {

this.radius = radius;

}

public double getArea() {

return Math.PI \* radius \* radius;

}

public double getPerimeter() {

return 2 \* Math.PI \* radius;

}

}

**Filename:** **Rectangle.** **java**

package shapes;

public class Rectangle {

private double length;

private double width;

public Rectangle(double length, double width) {

this.length = length;

this.width = width;

}

public double getArea() {

return length \* width;

}

public double getPerimeter() {

return 2 \* (length + width);

}

}

**RUN THIS🡺Filename:** **Pack\_prg.java**

import shapes.Circle;

import shapes.Rectangle;

class Pack\_prg {

public static void main(String[] args) {

Circle circle = new Circle(5.0);

Rectangle rectangle = new Rectangle(4.0, 6.0);

System.out.println("Circle:");

System.out.println("Area: " + circle.getArea());

System.out.println("Perimeter: " + circle.getPerimeter());

System.out.println("\nRectangle:");

System.out.println("Area: " + rectangle.getArea());

System.out.println("Perimeter: " + rectangle.getPerimeter());

}

}

**FILES**

**Filename:FileExample.java**

import java.io.FileWriter;

import java.io.FileReader;

import java.io.IOException;

public class FileExample {

public static void main(String[] args) {

try {

FileWriter writer = new FileWriter("students.txt");

writer.write("Name: Mephisto\n");

writer.write("Roll No: 202\n");

writer.write("Course: B.Sc. Computer Science\n");

writer.close();

System.out.println("Data successfully written to students.txt");

FileReader reader = new FileReader("students.txt");

int ch;

System.out.println("\n--- File Content ---");

while ((ch = reader.read()) != -1) {

System.out.print((char) ch);

}

reader.close();

}

catch (IOException e) {

System.out.println("An error occurred.");

e.printStackTrace(); } } }

**WORKING WITH FONTS AND COLORS**

**Filename:** **applet\_prg.java**

import java.applet.\*;

import java.awt.\*;

/\* <html>

<body>

<applet code="applet\_prg.class" width="400" height="150">

</applet>

</body>

</html>

\*/

public class applet\_prg extends Applet

{

public void paint(Graphics g) {

setBackground(Color.lightGray);

g.setColor(Color.red);

g.setFont(new Font("Arial", Font.BOLD, 20));

g.drawString("Arial Bold 20 in Red", 20, 30);

g.setColor(Color.blue);

g.setFont(new Font("Courier New", Font.ITALIC, 18));

g.drawString("Courier Italic 18 in Blue", 20, 60);

g.setColor(Color.green);

g.setFont(new Font("TimesRoman", Font.PLAIN, 22));

g.drawString("Times Roman Plain 22 in Green", 20, 90);

}

}

**COMMAND:🡺 1.javac applet\_prg.java**

**2.appletviewer applet\_prg.java**

**PARAMETER PASSING TECHNIQUE**

**Filename:** **ParameterApplet.java**

import java.applet.\*;

import java.awt.\*;

/\*

<applet code="ParameterApplet" width=300 height=200>

<param name="studentName" value="John Doe">

<param name="rollNo" value="101">

</applet>

\*/

public class ParameterApplet extends Applet {

String name;

String rollno;

public void init() {

name = getParameter("studentName");

rollno = getParameter("rollNo");

if (name == null) {

name = "Unknown";

}

if (rollno == null) {

rollno = "000";

}

}

public void paint(Graphics g) {

g.drawString("Student Details:", 20, 40);

g.drawString("Name: " + name, 20, 70);

g.drawString("Roll No: " + rollno, 20, 100);

}

}

**COMMAND🡺 javac ParameterApplet.java**

**appletviewer ParameterApplet.java**

**DRAWING DIFFERENT SHAPES**

**Filename:** **HumanFaceApplet.java**

import java.applet.Applet;

import java.awt.\*;

/\*

<html>

<body>

<applet code="HumanFaceApplet.class" width="350" height="400">

Your browser does not support Java Applets.

</applet>

</body>

</html>

\*/

public class HumanFaceApplet extends Applet {

public void paint(Graphics g) {

setBackground(Color.lightGray);

g.setColor(Color.white);

g.drawOval(50, 50, 200, 220);

g.setColor(Color.white);

g.fillOval(90, 110, 50, 30);

g.fillOval(180, 110, 50, 30);

g.setColor(Color.black);

g.fillOval(110, 115, 15, 15);

g.fillOval(200, 115, 15, 15);

g.drawLine(150, 150, 140, 190);

g.drawLine(150, 150, 160, 190);

g.drawLine(140, 190, 160, 190);

g.setColor(Color.red);

g.drawArc(110, 200, 90, 40, 180, 180);

}

}

**COMMAND🡺javac HumanFaceApplet.java**

**appletviewer HumanFaceApplet.java**