|  |  |
| --- | --- |
| **Program 1.1** | **Date: 01/01/2024** |
| **Write a Java program to check whether a string is palindrome or not.** | |

|  |
| --- |
| **CYCLE 1: Basic programs using datatypes, operators and control statement in java** |

**PROGRAM :**

|  |
| --- |
| import java.util.\*;  public class P7\_1\_palindrome {  public static void main(String args[]) {  Scanner s = new Scanner(System.in);  int i;  boolean flag = true;  try {  System.out.println("\nEnter a string to check for palindrome : ");  String str = s.nextLine();  str = str.toLowerCase(); // coverts entered string to lowercase  // Comparing one character at a time till middle of the string is reached  for (i = 0; i < (str.length() / 2); i++) {  if (str.charAt(i) != str.charAt(str.length() - i - 1)) {  flag = false;  break;  }  }  if (flag) {  System.out.println("\nstring " + str + " is Palindrome.");  } else {  System.out.println("\nstring " + str + " is Not Palindrome.");  }  } catch (Exception e) {  System.out.println("\nError : " + e);  }  }  } |

**OUTPUT :**

|  |
| --- |
| E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)\JAVA LAB>java P7\_1\_palindrome  Enter a string to check for palindrome :  malayalam  string malayalam is Palindrome.  E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)\JAVA LAB> |

|  |  |
| --- | --- |
| **Program 1.2** | **Date: 02/01/2024** |
| **Write a Java program to multiply two matrices.** | |

**PROGRAM :**

|  |
| --- |
| import java.util.\*;  public class P7\_2\_matrix\_multiply {  public static void main(String args[]) {  Scanner s = new Scanner(System.in);  int row1, col1, row2, col2;  int i, j;  try {  // MATRIX 1  System.out.println("\nEnter the number of rows of First matrix :");  row1 = s.nextInt();  System.out.println("\nEnter the number of columns of First matrix :");  col1 = s.nextInt();  System.out.println("\nEnter the Elements of First matrix : ");  int[][] matrix1 = new int[row1][col1];  for (i = 0; i < row1; i++) {  for (j = 0; j < col1; j++) {  matrix1[i][j] = s.nextInt();  }  }  // MATRIX 2  System.out.println("\nEnter the number of rows of Second matrix :");  row2 = s.nextInt();  System.out.println("\nEnter the number of columns of Second matrix :");  col2 = s.nextInt();  System.out.println("\nEnter the Elements of Second matrix : ");  int[][] matrix2 = new int[row2][col2];  for (i = 0; i < row2; i++) {  for (j = 0; j < col2; j++) {  matrix2[i][j] = s.nextInt();  }  }  // no.of columns of First matrix and no.of rows of Second matrix should be same  // for matrix multiplication posssible.  if (col1 != row2) {  System.out.println("\nMatrix multiplication is not possible !");  } else {  // RESULT MATRIX  int[][] result\_matrix = multiply\_matrix(matrix1, matrix2);  System.out.println("\nResult Matrix is --->");  for (i = 0; i < row1; i++) {  for (j = 0; j < col2; j++) {  System.out.print(result\_matrix[i][j] + "\t");  }  System.out.println("\n");  }  }  } catch (Exception e) {  System.out.println("\nError : " + e);  } finally {  s.close();  }  }  public static int[][] multiply\_matrix(int[][] matrix1, int[][] matrix2) {  int row1, col1, col2;  int i, j, k;  row1 = matrix1.length;  col2 = matrix2[0].length;  col1 = matrix1[0].length;  int[][] result = new int[row1][col2];  // multiplying  for (i = 0; i < row1; i++) {  for (j = 0; j < col2; j++) {  for (k = 0; k < col1; k++) {  result[i][j] += matrix1[i][k] \* matrix2[k][j];  }  }  }  return result;  }  } |

**OUTPUT :**

|  |
| --- |
| E:\MUHAMMAD ANSHAD P A\JAVA\JAVA LAB>java P7\_2\_matrix\_multiply  Enter the number of rows of First matrix :  2  Enter the number of columns of First matrix :  2  Enter the Elements of First matrix :  1  2  3  4  Enter the number of rows of Second matrix :  2  Enter the number of columns of Second matrix :  2  Enter the Elements of Second matrix :  4  5  6  7  Result Matrix is --->  16 19  36 43 |

|  |  |
| --- | --- |
| **Program 1.3** | **Date: 04/01/2024** |
| **Write a Java program to find the transpose of a matrix.** | |

**PROGRAM :**

|  |
| --- |
| import java.util.Scanner;  public class P7\_3\_transpose\_matrix {  public static void main(String[] args) {  Scanner scanner = new Scanner(System.in);  System.out.print("Enter the number of rows of the matrix: ");  int rows = scanner.nextInt();  System.out.print("Enter the number of columns of the matrix: ");  int cols = scanner.nextInt();  int[][] matrix = new int[rows][cols];  // Input matrix elements  System.out.println("Enter the elements of the matrix:");  for (int i = 0; i < rows; i++) {  for (int j = 0; j < cols; j++) {  matrix[i][j] = scanner.nextInt();  }  }  System.out.println("Original Matrix:");  printMatrix(matrix);  // Transpose the matrix  int[][] transposeMatrix = findTranspose(matrix);  System.out.println("Transpose of the Matrix:");  printMatrix(transposeMatrix);  scanner.close();  }  // Function to find the transpose of a matrix  public static int[][] findTranspose(int[][] matrix) {  int rows = matrix.length;  int cols = matrix[0].length;  int[][] transpose = new int[cols][rows];  for (int i = 0; i < rows; i++) {  for (int j = 0; j < cols; j++) {  transpose[j][i] = matrix[i][j];  }  }  return transpose;  }  // Function to print a matrix  public static void printMatrix(int[][] matrix) {  for (int[] row : matrix) {  for (int num : row) {  System.out.print(num + "\t");  }  System.out.println();  }  }  } |

**OUTPUT :**

|  |
| --- |
| PS E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)>  Enter the number of rows of the matrix: 2  Enter the number of columns of the matrix: 2  Enter the elements of the matrix:  4  5  6  7  Original Matrix:  4 5  6 7  Transpose of the Matrix:  4 6  5 7  PS E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)> |

|  |  |
| --- | --- |
| **Program 1.4** | **Date: 05/01/2024** |
| **Write a Java program to find the second smallest element in an array.** | |

**PROGRAM :**

|  |
| --- |
| import java.util.\*;  public class P7\_1\_palindrome {  public static void main(String args[]) {  Scanner s = new Scanner(System.in);  int i;  boolean flag = true;  try {  System.out.println("\nEnter a string to check for palindrome : ");  String str = s.nextLine();  str = str.toLowerCase(); // coverts entered string to lowercase  // Comparing one character at a time till middle of the string is reached  for (i = 0; i < (str.length() / 2); i++) {  if (str.charAt(i) != str.charAt(str.length() - i - 1)) {  flag = false;  break;  }  }  if (flag) {  System.out.println("\nstring " + str + " is Palindrome.");  } else {  System.out.println("\nstring " + str + " is Not Palindrome.");  }  } catch (Exception e) {  System.out.println("\nError : " + e);  }  }  } |

**OUTPUT :**

|  |
| --- |
| PS E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)>    Enter the number of elements in the array: 5  Enter the elements of the array:  4  3  7  8  1  The second smallest element in the array is: 3  PS E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)> |

|  |  |
| --- | --- |
| **Program 1.5** | **Date: 06/01/2024** |
| **Write a Java program to check whether a number is prime or not.** | |

**PROGRAM :**

|  |
| --- |
| import java.util.Scanner;  public class P7\_5\_prime {  public static void main(String[] args) {  Scanner scanner = new Scanner(System.in);  System.out.print("Enter a number to check if it's prime: ");  int number = scanner.nextInt();  if (isPrime(number)) {  System.out.println(number + " is a prime number.");  } else {  System.out.println(number + " is not a prime number.");  }  scanner.close();  }  public static boolean isPrime(int number) {  if (number <= 1) {  return false;  }  if (number <= 3) {  return true;  }  if (number % 2 == 0 || number % 3 == 0) {  return false;  }  for (int i = 5; i \* i <= number; i += 6) {  if (number % i == 0 || number % (i + 2) == 0) {  return false;  }  }  return true;  }  } |

**OUTPUT :**

|  |
| --- |
| PS E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)>  Enter a number to check if it's prime: 7  7 is a prime number.  PS E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)> |

|  |  |
| --- | --- |
| **Program 1.6** | **Date: 06/01/2024** |
| **Write a java program to demonstrate Bitwise logical operators, left shift and right shift operators.** | |

**PROGRAM :**

|  |
| --- |
| import java.util.\*;  public class P7\_6\_bitwise {  public static void main(String[] args) {  Scanner s = new Scanner(System.in);  try {  System.out.println("\nEnter the first integer : ");//Eg:12 -> Binary: 1100  int num1 = s.nextInt();  System.out.println("\nEnter the second integer : ");//Eg:7 -> Binary: 0111  int num2 = s.nextInt();  // Bitwise AND operator (&)  int resultAnd = num1 & num2; // Result: 4 (Binary: 0100)  System.out.println("Bitwise AND of " + num1 + " and " + num2 + " is: " + resultAnd);  // Bitwise OR operator (|)  int resultOr = num1 | num2; // Result: 15 (Binary: 1111)  System.out.println("Bitwise OR of " + num1 + " and " + num2 + " is: " + resultOr);  // Bitwise XOR operator (^)  int resultXor = num1 ^ num2; // Result: 11 (Binary: 1011)  System.out.println("Bitwise XOR of " + num1 + " and " + num2 + " is: " + resultXor);  // Bitwise NOT operator (~)  int resultNotNum1 = ~num1; // Result: -13 (Binary: 11111111 11111111 11111111 11110011)  System.out.println("Bitwise NOT of " + num1 + " is: " + resultNotNum1);  // Left shift operator (<<)  int resultLeftShift = num1 << 2; // Result: 48 (Binary: 110000)  System.out.println("Left shift of " + num1 + " by 2 is: " + resultLeftShift);  // Right shift operator (>>)  int resultRightShift = num2 >> 2; // Result: 1 (Binary: 0001)  System.out.println("Right shift of " + num2 + " by 2 is: " + resultRightShift);  } catch (Exception e) {  System.out.println("\nError : " + e);  }  s.close();  }  } |

**OUTPUT :**

|  |
| --- |
| PS E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)>  Enter the first integer :  12  Enter the second integer :  7  Bitwise AND of 12 and 7 is: 4  Bitwise OR of 12 and 7 is: 15  Bitwise XOR of 12 and 7 is: 11  Bitwise NOT of 12 is: -13  Left shift of 12 by 2 is: 48  Right shift of 7 by 2 is: 1  PS E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)> |

|  |  |
| --- | --- |
| **Program 1.7** | **Date: 08/01/2024** |
| **Write a java program to demonstrate Bitwise logical operators, left shift and right shift operators.** | |

**PROGRAM :**

|  |
| --- |
| import java.util.Scanner;  public class P7\_7\_roots\_of\_quadratic {  public static void main(String[] args) {  Scanner scanner = new Scanner(System.in);  System.out.println("Enter the coefficients of the quadratic equation (ax^2 + bx + c = 0):");  System.out.print("Enter the coefficient a: ");  double a = scanner.nextDouble();  System.out.print("Enter the coefficient b: ");  double b = scanner.nextDouble();  System.out.print("Enter the coefficient c: ");  double c = scanner.nextDouble();  double discriminant = b \* b - 4 \* a \* c;  if (discriminant > 0) {  double root1 = (-b + Math.sqrt(discriminant)) / (2 \* a);  double root2 = (-b - Math.sqrt(discriminant)) / (2 \* a);  System.out.println("Roots are real and different.");  System.out.println("Root 1 = " + root1);  System.out.println("Root 2 = " + root2);  } else if (discriminant == 0) {  double root = -b / (2 \* a);  System.out.println("Roots are real and equal.");  System.out.println("Root = " + root);  } else {  double realPart = -b / (2 \* a);  double imaginaryPart = Math.sqrt(-discriminant) / (2 \* a);  System.out.println("Roots are complex and different.");  System.out.println("Root 1 = " + realPart + " + " + imaginaryPart + "i");  System.out.println("Root 2 = " + realPart + " - " + imaginaryPart + "i");  }  scanner.close();  }  } |

**OUTPUT :**

|  |
| --- |
| PS E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)>  Enter the coefficients of the quadratic equation (ax^2 + bx + c = 0):  Enter the coefficient a: 1  Enter the coefficient b: -3  Enter the coefficient c: 2  Roots are real and different.  Root 1 = 2.0  Root 2 = 1.0  PS E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)> |

|  |
| --- |
| **CYCLE 2: Object Oriented Concepts** |

|  |  |
| --- | --- |
| **Program 2.1** | **Date: 16/01/2024** |
| **Write a Java program to calculate the area of different shapes namely circle, rectangle, trapezoid and triangle. (Use the concepts of JAVA like *this* keyword, constructor overloading and method overloading)** | |

**PROGRAM :**

|  |
| --- |
| import java.util.Scanner;  class AreaCalculator {  // Circle  public double calculateArea(double radius) {  return Math.PI \* radius \* radius;  }  // Rectangle  public double calculateArea(double length, double width) {  return length \* width;  }  // Trapezoid  public double calculateArea(double base1, double base2, double height) {  return (base1 + base2) \* height / 2;  }  // Triangle  public double calculateTriangleArea(double base, double height) {  return 0.5 \* base \* height;  }  }  public class P8\_1\_area\_of\_shapes {  public static void main(String[] args) {  Scanner scanner = new Scanner(System.in);  AreaCalculator areaCalculator = new AreaCalculator();  int choice = 0;  do {  System.out.println("\nChoose a shape to calculate its area:");  System.out.println("1. Circle");  System.out.println("2. Rectangle");  System.out.println("3. Trapezoid");  System.out.println("4. Triangle");  System.out.println("5. Exit.\nSelect any one : ");  choice = scanner.nextInt();  switch (choice) {  case 1:  System.out.println("Enter the radius of the circle:");  double radius = scanner.nextDouble();  System.out.println("Area of the circle: " + areaCalculator.calculateArea(radius));  break;  case 2:  System.out.println("Enter the length and width of the rectangle:");  double length = scanner.nextDouble();  double width = scanner.nextDouble();  System.out.println("Area of the rectangle: " + areaCalculator.calculateArea(length, width));  break;  case 3:  System.out.println("Enter the lengths of the two bases and the height of the trapezoid:");  double base1 = scanner.nextDouble();  double base2 = scanner.nextDouble();  double heightTrapezoid = scanner.nextDouble();  System.out.println(  "Area of the trapezoid: " + areaCalculator.calculateArea(base1, base2, heightTrapezoid));  break;  case 4:  System.out.println("Enter the base and height of the triangle:");  double baseTriangle = scanner.nextDouble();  double heightTriangle = scanner.nextDouble();  System.out.println("Area of the triangle: "  + areaCalculator.calculateTriangleArea(baseTriangle, heightTriangle));  break;  case 5:  System.out.println("Exiting....");  break;  // System.exit(0);  default:  System.out.println("Invalid choice!");  }  } while (choice != 5);  scanner.close();  }  } |

**OUTPUT :**

|  |
| --- |
| PS E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)>  Choose a shape to calculate its area:  1. Circle  2. Rectangle  3. Trapezoid  4. Triangle  5. Exit.  Select any one :  1  Enter the radius of the circle:  3  Area of the circle: 28.274333882308138  Choose a shape to calculate its area:  1. Circle  2. Rectangle  3. Trapezoid  4. Triangle  5. Exit.  Select any one :  2  Enter the length and width of the rectangle:  5  8  Area of the rectangle: 40.0  Choose a shape to calculate its area:  1. Circle  2. Rectangle  3. Trapezoid  4. Triangle  5. Exit.  Select any one :  3  Enter the lengths of the two bases and the height of the trapezoid:  5  6  7  Area of the trapezoid: 38.5  Choose a shape to calculate its area:  1. Circle  2. Rectangle  3. Trapezoid  4. Triangle  5. Exit.  Select any one :  4  Enter the base and height of the triangle:  7  4  Area of the triangle: 14.0  Choose a shape to calculate its area:  1. Circle  2. Rectangle  3. Trapezoid  4. Triangle  5. Exit.  Select any one :  5  Exiting....  PS E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)> |

|  |  |
| --- | --- |
| **Program 2.2** | **Date: 16/01/2024** |
| **Define a class called Rectangle with member variables length and width. Use appropriate member functions to calculate the perimeter and area of the rectangle. Define another member function *int sameArea(Rectangle)* that has one parameter of type Rectangle. *sameArea* returns 1 if the two Rectangles have the same area, and returns 0 if they don't. Use appropriate constructors to initialize the member variables(Use both default and parameterized constructor)** | |

**PROGRAM :**

|  |
| --- |
| import java.util.Scanner;  class Rectangle {  private double length;  private double width;  // Default constructor  public Rectangle() {  this.length = 0;  this.width = 0;  }  // Parameterized constructor  public Rectangle(double length, double width) {  this.length = length;  this.width = width;  }  // Getter methods  public double getLength() {  return length;  }  public double getWidth() {  return width;  }  // Setter methods  public void setLength(double length) {  this.length = length;  }  public void setWidth(double width) {  this.width = width;  }  // Method to calculate perimeter  public double calculatePerimeter() {  return 2 \* (length + width);  }  // Method to calculate area  public double calculateArea() {  return length \* width;  }  // Method to check if two rectangles have the same area  public int sameArea(Rectangle otherRectangle) {  if (this.calculateArea() == otherRectangle.calculateArea()) {  return 1;  } else {  return 0;  }  }  }  public class P8\_2\_member\_functions {  public static void main(String[] args) {  Scanner scanner = new Scanner(System.in);  // Taking input for first rectangle  System.out.println("Enter length and width for rectangle 1:");  double length1 = scanner.nextDouble();  double width1 = scanner.nextDouble();  Rectangle rectangle1 = new Rectangle(length1, width1);  // Taking input for second rectangle  System.out.println("Enter length and width for rectangle 2:");  double length2 = scanner.nextDouble();  double width2 = scanner.nextDouble();  Rectangle rectangle2 = new Rectangle(length2, width2);  // Calculating and printing perimeter and area of rectangle 1  System.out.println("Perimeter of rectangle 1: " + rectangle1.calculatePerimeter());  System.out.println("Area of rectangle 1: " + rectangle1.calculateArea());  // Calculating and printing perimeter and area of rectangle 2  System.out.println("Perimeter of rectangle 2: " + rectangle2.calculatePerimeter());  System.out.println("Area of rectangle 2: " + rectangle2.calculateArea());  // Checking if rectangle 1 and rectangle 2 have the same area  if (rectangle1.sameArea(rectangle2) == 1) {  System.out.println("Rectangle 1 and Rectangle 2 have the same area.");  } else {  System.out.println("Rectangle 1 and Rectangle 2 don't have the same area.");  }  scanner.close();  }  } |

**OUTPUT :**

|  |
| --- |
| E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)\JAVA LAB>java P8\_2\_member\_functions  Enter length and width for rectangle 1:  4  2  Enter length and width for rectangle 2:  4  2  Perimeter of rectangle 1: 12.0  Area of rectangle 1: 8.0  Perimeter of rectangle 2: 12.0  Area of rectangle 2: 8.0  Rectangle 1 and Rectangle 2 have the same area.  E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)\JAVA LAB> |

|  |  |
| --- | --- |
| **Program 2.3** | **Date: 16/01/2024** |
| **Write a main function to create two rectangle objects and display its area and perimeter. Check whether the two Rectangles have the same area and print a message indicating the result. (Use the concept of *this* pointer too)** | |

**PROGRAM :**

|  |
| --- |
| import java.util.Scanner;  class Rectangle {  private double length;  private double width;  // Default constructor  public Rectangle() {  this.length = 0;  this.width = 0;  }  // Parameterized constructor  public Rectangle(double length, double width) {  this.length = length;  this.width = width;  }  // Getter methods  public double getLength() {  return length;  }  public double getWidth() {  return width;  }  // Setter methods  public void setLength(double length) {  this.length = length;  }  public void setWidth(double width) {  this.width = width;  }  // Method to calculate perimeter  public double calculatePerimeter() {  return 2 \* (this.length + this.width); // using this pointer to refer to the instance variables  }  // Method to calculate area  public double calculateArea() {  return this.length \* this.width; // using this pointer to refer to the instance variables  }  // Method to check if two rectangles have the same area  public boolean sameArea(Rectangle otherRectangle) {  return this.calculateArea() == otherRectangle.calculateArea();  }  }  public class P8\_3\_two\_rectangleobjects {  public static void main(String[] args) {  Scanner scanner = new Scanner(System.in);  // Creating rectangle objects  Rectangle rectangle1 = new Rectangle();  Rectangle rectangle2 = new Rectangle();  // Taking input for rectangle 1  System.out.println("Enter length and width for rectangle 1:");  double length1 = scanner.nextDouble();  double width1 = scanner.nextDouble();  rectangle1.setLength(length1);  rectangle1.setWidth(width1);  // Taking input for rectangle 2  System.out.println("Enter length and width for rectangle 2:");  double length2 = scanner.nextDouble();  double width2 = scanner.nextDouble();  rectangle2.setLength(length2);  rectangle2.setWidth(width2);  // Displaying area and perimeter of rectangle 1  System.out.println("Area of rectangle 1: " + rectangle1.calculateArea());  System.out.println("Perimeter of rectangle 1: " + rectangle1.calculatePerimeter());  // Displaying area and perimeter of rectangle 2  System.out.println("Area of rectangle 2: " + rectangle2.calculateArea());  System.out.println("Perimeter of rectangle 2: " + rectangle2.calculatePerimeter());  // Checking if rectangle 1 and rectangle 2 have the same area  if (rectangle1.sameArea(rectangle2)) {  System.out.println("Rectangle 1 and Rectangle 2 have the same area.");  } else {  System.out.println("Rectangle 1 and Rectangle 2 don't have the same area.");  }  scanner.close();  }  } |

**OUTPUT :**

|  |
| --- |
| E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)\JAVA LAB>java P8\_3\_two\_rectangleobjects  Enter length and width for rectangle 1:  16  6  Enter length and width for rectangle 2:  16  6  Area of rectangle 1: 96.0  Perimeter of rectangle 1: 44.0  Area of rectangle 2: 96.0  Perimeter of rectangle 2: 44.0  Rectangle 1 and Rectangle 2 have the same area.  E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)\JAVA LAB> |

|  |  |
| --- | --- |
| **Program 2.4** | **Date: 17/01/2024** |
| **Write the definition for a class called Complex that has floating point data members for storing real and imaginary parts. Define a function *Complex sum(Complex)* to add two complex numbers** | |

**PROGRAM :**

|  |
| --- |
| import java.util.Scanner;  class Complex {  private double real;  private double imaginary;  // Default constructor  public Complex() {  this.real = 0;  this.imaginary = 0;  }  // Parameterized constructor  public Complex(double real, double imaginary) {  this.real = real;  this.imaginary = imaginary;  }  // Getter methods  public double getReal() {  return real;  }  public double getImaginary() {  return imaginary;  }  // Setter methods  public void setReal(double real) {  this.real = real;  }  public void setImaginary(double imaginary) {  this.imaginary = imaginary;  }  // Method to add two complex numbers  public Complex sum(Complex other) {  double realPart = this.real + other.getReal();  double imaginaryPart = this.imaginary + other.getImaginary();  return new Complex(realPart, imaginaryPart);  }  // Method to display complex number  public void display() {  System.out.println(this.real + " + " + this.imaginary + "i");  }  }  public class P8\_4\_complex {  public static void main(String[] args) {  Scanner scanner = new Scanner(System.in);  // Creating three complex number objects  Complex complex1 = new Complex();  Complex complex2 = new Complex();  Complex complex3;  // Taking input for complex number 1  System.out.println("Enter real and imaginary parts for complex number 1:");  double real1 = scanner.nextDouble();  double imaginary1 = scanner.nextDouble();  complex1.setReal(real1);  complex1.setImaginary(imaginary1);  // Taking input for complex number 2  System.out.println("Enter real and imaginary parts for complex number 2:");  double real2 = scanner.nextDouble();  double imaginary2 = scanner.nextDouble();  complex2.setReal(real2);  complex2.setImaginary(imaginary2);  // Calculating sum and assigning it to complex number 3  complex3 = complex1.sum(complex2);  // Displaying all complex numbers  System.out.println("Complex number 1:");  complex1.display();  System.out.println("Complex number 2:");  complex2.display();  System.out.println("Sum of complex number 1 and complex number 2:");  complex3.display();  scanner.close();  }  } |

**OUTPUT :**

|  |
| --- |
| E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)\JAVA LAB>java P8\_4\_complex  Enter real and imaginary parts for complex number 1:  3.5  2.7  Enter real and imaginary parts for complex number 2:  1.2  -4.5  Complex number 1:  3.5 + 2.7i  Complex number 2:  1.2 + -4.5i  Sum of complex number 1 and complex number 2:  4.7 + -1.7999999999999998i  E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)\JAVA LAB> |

|  |  |
| --- | --- |
| **Program 2.5** | **Date: 18/01/2024** |
| **Define a class called Time that has hours and minutes as integer. The class has the following member function: *Time sum(Time)* to sum two time object & return time a. Use the concept of constructor overloading to initialize the time**  **2.5.1** Write the definitions for each of the above member functions.  **2.5.2** Write main function to create three time objects. Set the value in two objects and call sum() to calculate sum and assign it in third object. Display all time objects. (Use the concept of *this* pointer too) | |

**PROGRAM :**

|  |
| --- |
| import java.util.Scanner;  class Time {  private int hours;  private int minutes;  // Default constructor  public Time() {  this.hours = 0;  this.minutes = 0;  }  // Parameterized constructor with hours and minutes  public Time(int hours, int minutes) {  this.hours = hours;  this.minutes = minutes;  }  // Getter methods  public int getHours() {  return hours;  }  public int getMinutes() {  return minutes;  }  // Setter methods  public void setHours(int hours) {  this.hours = hours;  }  public void setMinutes(int minutes) {  this.minutes = minutes;  }  // Method to sum two Time objects  public Time sum(Time other) {  int totalHours = this.hours + other.getHours();  int totalMinutes = this.minutes + other.getMinutes();  // Adjust minutes if they exceed 60  if (totalMinutes >= 60) {  totalHours += totalMinutes / 60;  totalMinutes %= 60;  }  return new Time(totalHours, totalMinutes);  }  // Method to display the time  public void display() {  System.out.println("Time: " + hours + " hours " + minutes + " minutes");  }  }  public class P8\_5\_time {  public static void main(String[] args) {  Scanner scanner = new Scanner(System.in);  // Creating three time objects  Time time1 = new Time();  Time time2 = new Time();  Time time3;  // Taking input for time object 1  System.out.println("Enter hours and minutes for time object 1:");  int hours1 = scanner.nextInt();  int minutes1 = scanner.nextInt();  time1.setHours(hours1);  time1.setMinutes(minutes1);  // Taking input for time object 2  System.out.println("Enter hours and minutes for time object 2:");  int hours2 = scanner.nextInt();  int minutes2 = scanner.nextInt();  time2.setHours(hours2);  time2.setMinutes(minutes2);  // Calculating sum and assigning it to time object 3  time3 = time1.sum(time2);  // Displaying all time objects  System.out.println("Time object 1:");  time1.display();  System.out.println("Time object 2:");  time2.display();  System.out.println("Sum of time object 1 and time object 2:");  time3.display();  scanner.close();  }  } |

**OUTPUT :**

|  |
| --- |
| E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)\JAVA LAB>java P8\_5\_time  Enter hours and minutes for time object 1:  6  35  Enter hours and minutes for time object 2:  7  45  Time object 1:  Time: 6 hours 35 minutes  Time object 2:  Time: 7 hours 45 minutes  Sum of time object 1 and time object 2:  Time: 14 hours 20 minutes  E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)\JAVA LAB> |

|  |  |
| --- | --- |
| **Program 2.5** | **Date: 19/01/2024** |
| **Create a class ‘Account’ with two overloaded constructors. The first constructor is used for initializing the name of account holder, the account number and the initial amount in the account. The second constructor is used for initializing the name of the account holder, the account number, the addresses, the type of account and the current balance. The Account class is having methods Deposit (), Withdraw (), and Get\_Balance(). Make the necessary assumption for data members and return types of the methods. Create objects of Account class and use them.** | |

**PROGRAM :**

|  |
| --- |
| import java.util.Scanner;  class Account {  private String accountHolderName;  private int accountNumber;  private String address;  private String accountType;  private double balance;  // First constructor  public Account(String accountHolderName, int accountNumber, double initialAmount) {  this.accountHolderName = accountHolderName;  this.accountNumber = accountNumber;  this.balance = initialAmount;  }  // Second constructor  public Account(String accountHolderName, int accountNumber, String address, String accountType, double currentBalance) {  this.accountHolderName = accountHolderName;  this.accountNumber = accountNumber;  this.address = address;  this.accountType = accountType;  this.balance = currentBalance;  }  // Method to deposit money  public void deposit(double amount) {  if (amount > 0) {  balance += amount;  System.out.println("Deposit successful. Current balance: " + balance);  } else {  System.out.println("Invalid deposit amount.");  }  }  // Method to withdraw money  public void withdraw(double amount) {  if (amount > 0 && amount <= balance) {  balance -= amount;  System.out.println("Withdrawal successful. Current balance: " + balance);  } else {  System.out.println("Insufficient funds or invalid withdrawal amount.");  }  }  // Method to get current balance  public double getBalance() {  return balance;  }  }  public class P8\_6\_account {  public static void main(String[] args) {  Scanner scanner = new Scanner(System.in);  // Account objects  Account account1 = null;  Account account2 = null;  // Taking input for account 1  System.out.println("Enter details for Account 1:");  System.out.print("Account holder name: ");  String name1 = scanner.nextLine();  System.out.print("Account number: ");  int number1 = scanner.nextInt();  System.out.print("Initial balance: ");  double initialBalance1 = scanner.nextDouble();  account1 = new Account(name1, number1, initialBalance1);  scanner.nextLine(); // Consume newline character  // Taking input for account 2  System.out.println("\nEnter details for Account 2:");  System.out.print("Account holder name: ");  String name2 = scanner.nextLine();  System.out.print("Account number: ");  int number2 = scanner.nextInt();  scanner.nextLine(); // Consume newline character  System.out.print("Address: ");  String address2 = scanner.nextLine();  System.out.print("Account type: ");  String type2 = scanner.nextLine();  System.out.print("Current balance: ");  double currentBalance2 = scanner.nextDouble();  account2 = new Account(name2, number2, address2, type2, currentBalance2);  // Menu-driven loop  boolean exit = false;  while (!exit) {  System.out.println("\nMenu:");  System.out.println("1. Deposit to Account 1");  System.out.println("2. Deposit to Account 2");  System.out.println("3. Withdraw from Account 1");  System.out.println("4. Withdraw from Account 2");  System.out.println("5. Exit");  System.out.print("Enter your choice: ");  int choice = scanner.nextInt();  switch (choice) {  case 1:  if (account1 != null) {  System.out.println("Enter deposit amount for account 1:");  double depositAmount1 = scanner.nextDouble();  account1.deposit(depositAmount1);  } else {  System.out.println("Account 1 not created yet.");  }  break;  case 2:  if (account2 != null) {  System.out.println("Enter deposit amount for account 2:");  double depositAmount2 = scanner.nextDouble();  account2.deposit(depositAmount2);  } else {  System.out.println("Account 2 not created yet.");  }  break;  case 3:  if (account1 != null) {  System.out.println("Enter withdrawal amount for account 1:");  double withdrawalAmount1 = scanner.nextDouble();  account1.withdraw(withdrawalAmount1);  } else {  System.out.println("Account 1 not created yet.");  }  break;  case 4:  if (account2 != null) {  System.out.println("Enter withdrawal amount for account 2:");  double withdrawalAmount2 = scanner.nextDouble();  account2.withdraw(withdrawalAmount2);  } else {  System.out.println("Account 2 not created yet.");  }  break;  case 5:  exit = true;  break;  default:  System.out.println("Invalid choice. Please enter a number between 1 and 5.");  }  }  scanner.close();  }  } |

**OUTPUT :**

|  |
| --- |
| E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)\JAVA LAB>java P8\_6\_account  Enter details for Account 1:  Account holder name: Anshad  Account number: 1001  Initial balance: 35000  Enter details for Account 2:  Account holder name: Nihal  Account number: 1005  Address: Nihal house,Kannur  Account type: Savings  Current balance: 55000  Menu:  1. Deposit to Account 1  2. Deposit to Account 2  3. Withdraw from Account 1  4. Withdraw from Account 2  5. Exit  Enter your choice: 1  Enter deposit amount for account 1:  25000  Deposit successful. Current balance: 60000.0  Menu:  1. Deposit to Account 1  2. Deposit to Account 2  3. Withdraw from Account 1  4. Withdraw from Account 2  5. Exit  Enter your choice: 2  Enter deposit amount for account 2:  7000  Deposit successful. Current balance: 62000.0  Menu:  1. Deposit to Account 1  2. Deposit to Account 2  3. Withdraw from Account 1  4. Withdraw from Account 2  5. Exit  Enter your choice: 3  Enter withdrawal amount for account 1:  6500  Withdrawal successful. Current balance: 53500.0  Menu:  1. Deposit to Account 1  2. Deposit to Account 2  3. Withdraw from Account 1  4. Withdraw from Account 2  5. Exit  Enter your choice: 4  Enter withdrawal amount for account 2:  100  Withdrawal successful. Current balance: 61900.0  Menu:  1. Deposit to Account 1  2. Deposit to Account 2  3. Withdraw from Account 1  4. Withdraw from Account 2  5. Exit  Enter your choice: 5  E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)\JAVA LAB> |

|  |
| --- |
| **CYCLE 3: Inheritance, method overloading and overriding, Polymorphism** |

|  |  |
| --- | --- |
| **Program 3.1** | **Date: 20/01/2024** |
| **Write a Java program which creates a class named 'Employee' having the following members: Name, Age, Phone number, Address, Salary. It also has a method named 'print- Salary( )' which prints the salary of the Employee. Two classes 'Officer' and 'Manager' inherits the 'Employee' class. The 'Officer' and 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign name, age, phone number, address and salary to an officer and a manager by making an object of both of these classes and print the same.** | |

**PROGRAM :**

|  |
| --- |
| import java.util.Scanner; |

**OUTPUT :**

|  |
| --- |
| E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)\JAVA LAB>java P8\_5\_time |

|  |  |
| --- | --- |
| **Program 2.5** | **Date: 18/01/2024** |
| **De** | |

**PROGRAM :**

|  |
| --- |
| import java.util.Scanner; |

**OUTPUT :**

|  |
| --- |
| E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)\JAVA LAB>java P8\_5\_time |

|  |  |
| --- | --- |
| **Program 2.5** | **Date: 18/01/2024** |
| **De** | |

**PROGRAM :**

|  |
| --- |
| import java.util.Scanner; |

**OUTPUT :**

|  |
| --- |
| E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)\JAVA LAB>java P8\_5\_time |

|  |  |
| --- | --- |
| **Program 2.5** | **Date: 18/01/2024** |
| **De** | |

**PROGRAM :**

|  |
| --- |
| import java.util.Scanner; |

**OUTPUT :**

|  |
| --- |
| E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)\JAVA LAB>java P8\_5\_time |

|  |  |
| --- | --- |
| **Program 2.5** | **Date: 18/01/2024** |
| **De** | |

**PROGRAM :**

|  |
| --- |
| import java.util.Scanner; |

**OUTPUT :**

|  |
| --- |
| E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)\JAVA LAB>java P8\_5\_time |

|  |  |
| --- | --- |
| **Program 2.5** | **Date: 18/01/2024** |
| **De** | |

**PROGRAM :**

|  |
| --- |
| import java.util.Scanner; |

**OUTPUT :**

|  |
| --- |
| E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)\JAVA LAB>java P8\_5\_time |

|  |  |
| --- | --- |
| **Program 2.5** | **Date: 18/01/2024** |
| **De** | |

**PROGRAM :**

|  |
| --- |
| import java.util.Scanner; |

**OUTPUT :**

|  |
| --- |
| E:\MCA\SEM 2\JAVA PROGRAMMING (MCA202)\JAVA LAB>java P8\_5\_time |