Java Programming [CSE201] Enrolment No.: 23DCS056

CHAROTAR UNIVERSITY OF SCIENCE & TECHNOLOGY

**DEVANG PATEL INSTITUTE OF ADVANCE TECHNOLOGY & RESEARCH**

Department of Computer Science & Engineering

Subject Name: Java Programming

Semester: 3

Subject Code: 203

Academic year: 2024-2025

Part - 3

|  |  |
| --- | --- |
| **No.** | **Aim of the Practical** |
| 12 | Imagine you are developing a currency conversion tool for a travel agency. This tool should be able to convert an amount in Pounds to Rupees. For simplicity, we assume the conversion rate is fixed: 1 Pound = 100 Rupees. The tool should be able to take input both from command-line arguments and interactively from the user.  **PROGRAM:**  import java.util.\*;  public class Practical12 {  public static void main(String[] args) {  Scanner sc = new Scanner(System.in);    float pound,rupee;  System.out.println("enter the amount in pounds:");    pound= sc.nextFloat();    rupee=100\*pound;    System.out.println("amount in rupees is "+rupee);  sc.close();      }  }  **OUTPUT:**    **CONCLUSION:**  In this experiment, we learnt how to perform mathematical manipulations on floating and integer values and learnt about implicit type conversion. |
| 13 | Create a class called Employee that includes three pieces of information as instance variables—a first name (type String), a last name (type String) and a monthly salary (double). Your class should have a constructor that initializes the three instance variables. Provide a set and a get method for each instance variable. If the monthly salary is not positive, set it to 0.0. Write a test application named EmployeeTest that demonstrates class Employee’s capabilities. Create two Employee objects and display each object’s yearly salary. Then give each Employee a 10% raise and display each Employee’s yearly salary  again.  **PROGRAM:**  class Employee {  private String firstName;  private String lastName;  private double monthlySalary;    // Constructor to initialize the instance variables  public Employee(String firstName, String lastName, double monthlySalary) {  this.firstName = firstName;  this.lastName = lastName;  if (monthlySalary > 0) {  this.monthlySalary = monthlySalary;  } else {  this.monthlySalary = 0.0;  System.out.println("Invalid salary entered! Employee salary set to 0.0");  }  }    // Getter and setter for first name  public String getFirstName() {  return firstName;  }  public void setFirstName(String firstName) {  this.firstName = firstName;  }    // Getter and setter for last name  public String getLastName() {  return lastName;  }  public void setLastName(String lastName) {  this.lastName = lastName;  }    // Getter and setter for monthly salary  public double getMonthlySalary() {  return monthlySalary;  }  public void setMonthlySalary(double monthlySalary) {  if (monthlySalary > 0) {  this.monthlySalary = monthlySalary;  } else {  this.monthlySalary = 0.0;  System.out.println("Invalid salary entered! Employee salary set to 0.0");  }  }  }  public class Practical13 {  public static void main(String[] args) {  // Creating two Employee objects  Employee emp1 = new Employee("Vansh", "Malani", 3000);  Employee emp2 = new Employee("James", "Dhandhukiya", 4000);  // Displaying yearly salary for each employee  System.out.println(emp1.getFirstName() + " " + emp1.getLastName() + "'s yearly salary: " + (emp1.getMonthlySalary() \* 12));  System.out.println(emp2.getFirstName() + " " + emp2.getLastName() + "'s yearly salary: " + (emp2.getMonthlySalary() \* 12));  // Giving each employee a 10% raise  emp1.setMonthlySalary(emp1.getMonthlySalary() \* 1.10);  emp2.setMonthlySalary(emp2.getMonthlySalary() \* 1.10);  // Displaying yearly salary again after the raise  System.out.println("After 10% raise:");  System.out.println(emp1.getFirstName() + " " + emp1.getLastName() + "'s yearly salary: " + (emp1.getMonthlySalary() \* 12));  System.out.println(emp2.getFirstName() + " " + emp2.getLastName() + "'s yearly salary: " + (emp2.getMonthlySalary() \* 12));  }  }  **OUTPUT:**    **CONCLUSION:**  In this experiment, we learnt how to define classes and initialize the objects of that class and use them for storing and fetching data and manipulating the data such as incrementing and displaying it. |
| 14 | Create a class called Date that includes three pieces of information as instance variables—a month (type int), a day (type int) and a year (type int). Your class should have a constructor that initializes the three instance variables and assumes that the values provided are correct. Provide a set and a get method for each instance variable. Provide a method displayDate that displays the month, day and year separated by forward slashes (/). Write a test application named DateTest that demonstrates class Date’s capabilities.  **PROGRAM:**  public class Practical14 {  public static void main(String[] args) {  DateTime d1 = new DateTime(10, 7, 2005);  d1.getDate();  }  }  class DateTime {  int day, month, year;  public DateTime(int d, int m, int y) {  this.day = d;  this.month = m;  this.year = y;  System.out.println("Date initialized successfully!");  }  public void setDate(int d, int m, int y) {  this.day = d;  this.month = m;  this.year = y;  }  public void getDate() {  System.out.println(this.month+"/"+this.day+"/"+this.year);  }  }  **OUTPUT:**    **CONCLUSION:**  In this experiment we learnt how to create classes with default constructor and make getter and setter methods to access and display data. |
| 15 | Write a program to print the area of a rectangle by creating a class named 'Area' taking the values of its length and breadth as parameters of its constructor and having a method named 'return Area' which returns the area of the rectangle. Length and breadth of rectangle are entered through keyboard.  **PROGRAM:**  public class Practical15 {  public static void main(String[] args) {  Area a1 = new Area(10, 20);  Area a2 = new Area(40, 50);    System.out.println("the area of Rectangle 1 is:" + a1.returnArea());    System.out.println("the area of Rectangle 2 is:" + a2.returnArea());  }  }  class Area {  int length, breadth;  public Area(int l, int b) {  this.length = l;  this.breadth = b;  }  public float returnArea() {  float area;  area = length \* breadth;  return area;  }  }  **OUTPUT:**    **CONCLUSION:**  This experiment taught us how we can use classes to solve mensuration problems and find area of a given rectangle by just its breadth and length. |
| 15 | **Supplementary Experiment:**  1.Write a Java program to create a class called "Airplane" with a flight number, destination, and departure time attributes, and methods to check flight status and delay.  **PROGRAM:**  import java.util.\*;  public class Practical15sup {  public static void main(String[] args) {  Airplane a1 = new Airplane(37, "3 AM", "Delhi", 0);  Airplane a2 = new Airplane(7, "6 AM", "Mumbai", 1);  a1.FLightStat();  a1.CheckDelay();  a2.FLightStat();  a2.CheckDelay();  }  }  class Airplane {  int flightno;  String departure, destination;  int flag;  Random rand = new Random();  int ran;  public Airplane(int f, String dep, String des,int flag) {  this.flightno = f;  this.departure = dep;  this.destination = des;  this.flag = flag;  if (this.flag == 0) {  ran = rand.nextInt(50);  }  else {  ran = 500;  }  }  public void FLightStat() {  if (this.flag == 0) {  System.out.println("Flight is on schedule and arriving in " + this.ran + " minutes");  } else {  this.ran = this.ran / 60;  System.out.println("Flight is delayed and arriving in " + ran + " hours");  }  }  public void CheckDelay() {  if (flag == 0) {  System.out.println("there is no delay");  }  else {  this.ran = this.ran / 60;  System.out.println("there is a delay. The flight will arrive in "+ran+" hours");  }    }  }  **OUTPUT:**    **CONCLUSION:**  In this program we learnt how to create methods for different objects and I learnt to use the random class for generating a random value for the delay function. |
| 16 | Print the sum, difference and product of two complex numbers by creating a class named ‘Complex’ with separate methods for each operation whose real and imaginary parts are entered by user.  **PROGRAM:**  public class Practical16 {  public static void main(String[] args) {  complex c1, c2,c3,c4,c5;  c1= new complex(10,5);  c2=new complex(5,20);  c3 = new complex();  c4 = new complex();  c5 = new complex();  c3.add(c1,c2);  System.out.println("the value of c1+c2 is ");  c3.display();  c4.diff(c1, c2);  System.out.println("the value of c1-c2 is ");  c4.display();  c5.product(c1, c2);  System.out.println("the value of c1\*c2 is ");  c5.display();    }  }  class complex {  int real;  int imaginary;  public complex(int r, int c) {  this.real = r;  this.imaginary = c;  }  public complex() {  this.real = 0;  this.imaginary = 0;  }    public void add(complex c1, complex c2) {  this.real = c1.real + c2.real;  this.imaginary = c1.imaginary + c2.imaginary;  }    public void diff(complex c1, complex c2) {  this.real = c1.real - c2.real;  this.imaginary = c1.imaginary - c2.imaginary;  }    public void product(complex c1, complex c2) {  this.real = c1.real \* c2.real - c1.imaginary \* c2.imaginary;  this.imaginary = c1.real \* c2.imaginary + c2.real \* c1.imaginary;  }    public void display() {  System.out.println(this.real+" + "+this.imaginary+"i");  }  }  **OUTPUT:**    **CONCLUSION:**  **In this program we learnt how to create constructors for complex numbers operations such as addition, subtraction and multiplication.** |