Java Programming [CSE201] Enrolment No.:23DCS100

CHAROTAR UNIVERSITY OF SCIENCE & TECHNOLOGY

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

Department of Computer Science & Engineering

Subject Name: Java Programming

Semester: III

Subject Code: CSE201

Academic year: 2024-25

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 CODE:**  import *java.util.Scanner*;  *public* *class* CurrencyConverter {  *private* *static* *final* double CONVERSION\_RATE = 100.0;  *public* *static* void main(String[] args) {          if (args.length > 0) {              double pounds = Double.parseDouble(args[0]);              double rupees = convertToRupees(pounds);              System.out.printf("%.2f Pounds = %.2f Rupees\n", pounds, rupees);          } else {              Scanner scanner = new Scanner(System.in);              System.out.print("Enter amount in Pounds: ");              double pounds = scanner.nextDouble();              double rupees = convertToRupees(pounds);              System.out.printf("%.2f Pounds = %.2f Rupees\n", pounds, rupees);          }      }  *private* *static* double convertToRupees(double pounds) {          return pounds \* CONVERSION\_RATE;      }  }  **OUTPUT:**    **CONCLUSION:**  This Java program defines a `CurrencyConverter` that converts an amount from Pounds to Rupees using a fixed conversion rate of 1 Pound = 100 Rupees. It takes input either from command-line arguments or interactively from the user. The program handles invalid input gracefully, ensuring that only valid numerical values are processed for 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 CODE:**  class Employee {  private String firstName;  private String lastName;  private int monthlySalary;  public Employee(String firstName, String lastName, int monthlySalary) {  this.firstName = firstName;  this.lastName = lastName;  this.monthlySalary = (monthlySalary > 0) ? monthlySalary : 0;  }  public String getFirstName() {  return firstName;  }  public void setFirstName(String firstName) {  this.firstName = firstName;  }  public String getLastName() {  return lastName;  }  public void setLastName(String lastName) {  this.lastName = lastName;  }  public int getMonthlySalary() {  return monthlySalary;  }  public void setMonthlySalary(int monthlySalary) {  this.monthlySalary = (monthlySalary > 0) ? monthlySalary : 0;  }  public int getYearlySalary() {  return monthlySalary \* 12;  }  public void applyRaise() {  this.monthlySalary \*= 1.10;  }  }  public class EmployeeTest {  public static void main(String[] args) {  Employee employee1 = new Employee("Daksh","Prajapati", 3000);  Employee employee2 = new Employee("Harsh", "Patel", 3500);  System.out.println("Employee 1: " + employee1.getFirstName() + " " + employee1.getLastName());  System.out.println("Yearly Salary: $" + employee1.getYearlySalary());  System.out.println("Employee 2: " + employee2.getFirstName() + " " + employee2.getLastName());  System.out.println("Yearly Salary: $" + employee2.getYearlySalary());  employee1.applyRaise();  employee2.applyRaise();  System.out.println("\nAfter 10% raise:");  System.out.println("Employee 1: " + employee1.getFirstName() + " " + employee1.getLastName());  System.out.println("Yearly Salary: $" + employee1.getYearlySalary());  System.out.println("Employee 2: " + employee2.getFirstName() + " " + employee2.getLastName());  System.out.println("Yearly Salary: $" + employee2.getYearlySalary());  }  }  **OUTPUT:**    **CONCLUSION:**  This Java program defines an `Employee` class with attributes for first name, last name, and monthly salary, including methods to get and set these attributes, calculate yearly salary, and apply a 10% raise. The `EmployeeTest` class demonstrates the functionality by creating two `Employee` objects, displaying their yearly salaries, applying a 10% raise, and then displaying the updated salaries. |
| 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 CODE:**  class Date{  private int month;  private int day;  private int year;  Date(){  }  Date(int month, int day,int year){  this.month=month;  this.day=day;  this.year=year;  }  public int getmonth() {  return month;  }  public void setmonth(int month) {  this.month = month;  }  public int getday() {  return day;  }  public void setday(int day) {  this.day = day;  }  public int getyear() {  return year;  }  public void setyear(int year) {  this.year = year;  }  void displaydata(){  if(this.month>12 ){  System.out.println("Invalid month");  }  else if(this.month==1 || (this.month>=3 && this.month<=12)){  System.out.println(this.month+"/"+this.day+"/"+this.year);  }  else{  if((this.day==28 && this.year%4==0 && this.month==2) || (this.day>29 && this.month==2))  System.out.println(this.month+"/"+this.day+"/"+this.year);  }  }  }  public class DateTest{  public static void main(String[] args) {  Date d1=new Date(12,12,2024);  d1.displaydata();  }  }  **OUTPUT:**    **CONCLUSION:**  This Java program defines a `Date` class to represent a date with month, day, and year attributes. It includes getter and setter methods for each attribute and a `displaydata` method to print the date in a "MM/DD/YYYY" format, validating the month and handling leap year logic for February. The `DateTest` class demonstrates the capabilities of the `Date` class by creating a `Date` object and displaying it. |
| 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 'returnArea' which returns the area of the rectangle. Length and breadth of rectangle are entered through keyboard.  **PROGRAMCODE:**  import java.util.Scanner;  class Area{  private double length;  private double breadth;  Area(){  }  Area(double length,double breadth){  this.length=length;  this.breadth=breadth;  }  public void display(){  System.out.printf("Area of reactangle is : "+(double)Math.round((this.length\*this.breadth)\*100)/100);  }  }  public class JAVA\_P15 {  public static void main(String[] args) {  double a,b;  Scanner sc =new Scanner(System.in);  System.out.print("Enter the length : ");  a=sc.nextDouble();  System.out.print("Enter the breadth : ");  b=sc.nextDouble();  Area a1=new Area(a,b);  a1.display();  }  }  **OUTPUT:**  **CONCLUSION :**  This Java program defines an `Area` class to represent a rectangle with length and breadth attributes. It includes a constructor for initializing these attributes and a `display` method to calculate and print the area of the rectangle, rounded to two decimal places. The `JAVA\_P15` class demonstrates the `Area` class by taking user input for length and breadth, creating an `Area` object, and displaying the calculated area. |
| 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 CODE :**  import java.util.Scanner;  class complex{  private float real;  private float imaginary;  complex(){  }  complex(float real,float imaginary){  this.real=real;  this.imaginary=imaginary;  }  public complex(complex other) {  this.real = other.real;  this.imaginary = other.imaginary;  }  public float getreal() {  return real;  }  public void setreal(float real) {  this.real = real;  }  public float getimaginary() {  return imaginary;  }  public void setlength(float imaginary) {  this.imaginary = imaginary;  }  public complex add(complex other){  complex temp =new complex();  temp.real=this.real+other.real;  temp.imaginary=this.imaginary+other.imaginary;  return (temp);  }  public complex substraction(complex other){  complex temp =new complex();  temp.real=this.real-other.real;  temp.imaginary=this.imaginary-other.imaginary;  return (temp);  }  public complex multiplication(complex other){  complex temp =new complex();  temp.real=this.real \* other.real - this.imaginary \* other.imaginary;  temp.imaginary=this.real \* other.imaginary + this.imaginary \* other.real;  return (temp);  }  public complex divide(complex other){  complex temp =new complex();  float denominator = other.real \* other.real + other.imaginary \* other.imaginary;  temp.real=(this.real \* other.real + this.imaginary \* other.imaginary) / denominator;  temp.imaginary=(this.imaginary \* other.real - this.real \* other.imaginary) / denominator;  return (temp);  }  public void display(){  System.out.println(real+"+"+imaginary+"i");  }  }  public class JAVA\_P16 {  public static void main(String[] args) {  float real1,imaginary1;  float real2,imaginary2;  int x;  Scanner sc=new Scanner(System.in);  System.out.print("Enter the real part of A : ");  real1=sc.nextFloat();  System.out.print("Enter the imaginary part of A : ");  imaginary1=sc.nextFloat();  System.out.print("Enter the real part of B : ");  real2=sc.nextFloat();  System.out.print("Enter the imaginary part of B : ");  imaginary2=sc.nextFloat();  complex c1= new complex(real1,imaginary1);  complex c2= new complex(real2,imaginary2);  complex c3;  System.out.println("1)Addition\n2)Subtraction\n3)Multiplication\n4)Division\n");  System.out.print("Enter your choice : ");  x=sc.nextInt();  switch (x) {  case 1:  c3=c1.add(c2);  c3.display();  break;  case 2:  c3=c1.substraction(c2);  c3.display();  break;  case 3:  c3=c1.multiplication(c2);  c3.display();  break;  case 4:  c3=c1.divide(c2);  c3.display();  break;  default:  }  }  }  **OUTPUT:**    **CONCLUDION:**  This Java program defines a `complex` class to represent complex numbers and perform operations such as addition, subtraction, multiplication, and division. It uses a main method to take user input for two complex numbers and then allows the user to select an operation to perform on these numbers, displaying the result. |