Questions:

1. Simple Calculator: Create a program that prompts the user to enter two numbers and perform basic arithmetic operations such as addition, subtraction, multiplication, and division. Use variables to store the user inputs and the results of the calculations.

2. Temperature Conversion: Build a program that converts temperature values between Celsius and Fahrenheit. Prompt the user to enter a temperature in one unit and display the converted value in the other unit. Utilize variables to store the input and output values.

3. Circle Properties: Design a program that calculates and displays various properties of a circle, such as its radius, diameter, circumference, and area. Prompt the user to enter the radius, and use variables to store the calculations.

4. Grade Calculator: Develop a program that calculates the average grade of a student based on individual test scores. Prompt the user to enter the number of tests and the corresponding scores. Use variables to store the input values and calculate the average.

5. BMI Calculator: Create a program that calculates the Body Mass Index (BMI) based on user input for weight and height. Display the calculated BMI along with an appropriate message indicating the weight category. Utilize variables to store the weight, height, and BMI values.

6. Rectangle Area and Perimeter: Build a program that prompts the user to enter the length and width of a rectangle. Calculate and display the area and perimeter of the rectangle using variables to store the input and calculated values.

7. Currency Conversion: Design a program that converts an amount of money from one currency to another. Prompt the user to enter the amount and the exchange rate. Use variables to store the input values and the converted amount.

8. Time Conversion: Develop a program that converts time from seconds to hours, minutes, and seconds. Prompt the user to enter the time in seconds, and use variables to store the calculated values for hours, minutes, and remaining seconds.

9. Create a Class Hierarchy: Define a class hierarchy representing different types of vehicles. Start with a base class "Vehicle" and create derived classes such as "Car," "Motorcycle," and "Truck." Add relevant attributes and methods to each class that capture their unique characteristics.

10. Implement Polymorphism: Extend the vehicle hierarchy assignment by implementing polymorphic behavior. Create a method, such as "drive," in the base class, and override it in the derived classes toprovide specific implementations. Demonstrate polymorphic behavior by invoking the method on different objects of the derived classes.

11. Shape Inheritance: Create a class hierarchy representing different shapes, such as "Circle," "Rectangle," and "Triangle." Each shape should have common attributes like area and perimeter, as well as specific attributes and methods. Implement inheritance and polymorphism to calculate and display the area and perimeter of different shapes.

12. Employee Inheritance: Design an employee management system using inheritance. Create a base class "Employee" and derived classes such as "Manager," "Developer," and "Tester." Each derived class should have specific attributes and methods, along with common attributes like name and salary. Implement functionality to calculate the total salary of all employees.

13. Bank Account Inheritance: Create a class hierarchy representing different types of bank accounts, such as "SavingsAccount" and "CurrentAccount." Implement common functionalities like deposit and withdrawal in the base class and specific functionalities like interest calculation in the derived classes.

14. Animal Inheritance: Design an animal hierarchy with a base class "Animal" and derived classes such as "Cat," "Dog," and "Bird." Add appropriate methods and attributes to represent different behaviors and characteristics of each animal. Implement polymorphism by invoking common methods on objects of different animal types.

15. School System Inheritance: Model a school system using inheritance. Create a base class "Person" and derived classes such as "Teacher" and "Student." Implement functionalities like assigning subjects to teachers and registering students for courses. Demonstrate inheritance and polymorphism by accessing and displaying information about different types of people in the school system.

1). package com.bov.assignment1;

/\* Simple Calculator: Create a program that prompts the user to enter two numbers and perform basic arithmetic

operations such as addition, subtraction, multiplication, and division. Use variables to store the user

inputs and the results of the calculations.\*/

import java.util.Scanner;

class SimpleCalculator

{

public static void main(String args[])

{

int a;

int b;

int add;

int sub;

int mul;

float div;

int choice;

Scanner sc = new Scanner(System.in);

System.out.println("Enter a value : ");

a = sc.nextInt();

System.out.println("Enter b value : ");

b = sc.nextInt();

System.out.println("Enter choice of 1. Add \n 2. Subtract \n 3. Multuply \n 4. Division \n Enter choice : ");

choice = sc.nextInt();

switch(choice)

{

case 1:

add = a+b;

System.out.println("Addition of "+a+" and "+b+" is : " +add);

break;

case 2:

sub = a-b;

System.out.println("Subtraction of "+a+" and "+b+" is : "+sub);

break;

case 3:

mul = a\*b;

System.out.println("Multiplication of "+a+" and "+b+" is : "+mul);

break;

case 4:

div = a/b;

System.out.println("Division of "+a+" and "+b+" is : "+div);

break;

default:

System.out.println("You have entered incorrectly");

}

}

}

Output:

Enter a value :

5

Enter b value :

6

Enter choice of 1. Add

2. Subtract

3. Multuply

4. Division

Enter choice :

1

Addition of 5 and 6 is : 11

2). package com.bov.assignment1;

/\*Temperature Conversion: Build a program that converts temperature values between Celsius and Fahrenheit.

Prompt the user to enter a temperature in one unit and display the converted value in the other unit.

Utilize variables to store the input and output values. \*/

import java.util.Scanner;

class TemperatureConversion

{

public static void main(String []args)

{

float celsius;

float fahrenheit;

double result;

int choice;

Scanner sc = new Scanner(System.in);

System.out.println("Enter from below - \n 1. Celsius to fahrenheit \n 2. Fahrenheit to Celsius \n Enter choice : ");

choice = sc.nextInt();

switch(choice)

{

case 1:

System.out.println("Enter temperature in celsius : ");

celsius = sc.nextFloat();

result = 1.8 \* celsius + 32;

System.out.println("The temperature "+celsius+" C in fahrenheit is : "+result);

break;

case 2:

System.out.println("Enter temperature in fahrenheit : ");

fahrenheit = sc.nextFloat();

result = (fahrenheit-32)/1.8;

System.out.println("Temperature "+fahrenheit+" F in celsius is : "+result);

break;

default:

System.out.println("You have entered incorrectly");

}

}

}

Output:

Enter from below -

1. Celsius to fahrenheit

2. Fahrenheit to Celsius

Enter choice :

1

Enter temperature in celsius :

37

The temperature 37.0 C in fahrenheit is : 98.60000000000001

3). package com.bov.assignment1;

/\* Circle Properties: Design a program that calculates and displays various properties of a circle,

such as its radius, diameter, circumference, and area. Prompt the user to enter the radius,

and use variables to store the calculations. \*/

import java.util.Scanner;

class CircleProperties

{

public static void main(String args[])

{

float radius;

float diameter;

double circumference;

double area;

Scanner sc = new Scanner(System.in);

System.out.println("Enter radius of the circle : ");

radius = sc.nextFloat();

diameter = radius \* 2;

circumference = 2 \* 3.14 \* radius;

area = 3.14 \* radius \* radius;

System.out.println("Diameter of the circle is : "+diameter);

System.out.println("Circumference of the circle is : "+circumference);

System.out.println("Area of the circle is : "+area);

}

}

Output:

Enter radius of the circle :

5

Diameter of the circle is : 10.0

Circumference of the circle is : 31.400000000000002

Area of the circle is : 78.5

4). package com.bov.assignment1;

/\* Grade Calculator: Develop a program that calculates the average grade of a student based on individual test scores.

Prompt the user to enter the number of tests and the corresponding scores.

Use variables to store the input values and calculate the average. \*/

import java.util.Scanner;

class GradeCalculator

{

public static void main(String args[])

{

int nt;

int total = 0;

float average;

Scanner sc = new Scanner(System.in);

System.out.println("Enter no. of test scores : ");

nt = sc.nextInt();

int testScore[] = new int[nt];

testScore = new int[nt];

for(int i=0; i<nt; i++)

{

System.out.println("Enter test score "+(i+1));

testScore[i] = sc.nextInt();

total = total + testScore[i];

}

average = total/nt;

System.out.println("The total score is : "+total+" and average score is : "+average);

if(average >=80)

{

System.out.println("The grade is : Distinction");

}

else if(average <80 && average >=60)

{

System.out.println("The grade is : A-Grade");

}

else if(average <60 && average >=50)

{

System.out.println("The grade is : B-Grade");

}

else if(average <50 && average >= 40)

{

System.out.println("The Grade is : C-Grade");

}

else

{

System.out.println("You have not qualified the exam");

}

}

}

Output:

Enter no. of test scores :

5

Enter test score 1

86

Enter test score 2

75

Enter test score 3

70

Enter test score 4

96

Enter test score 5

93

The total score is : 420 and average score is : 84.0

The grade is : Distinction

5). package com.bov.assignment1;

/\* BMI Calculator: Create a program that calculates the Body Mass Index (BMI) based on user input for weight and height.

Display the calculated BMI along with an appropriate message indicating the weight category.

Utilize variables to store the weight, height, and BMI values. \*/

import java.util.Scanner;

class BmiCalculator

{

public static void main(String args[])

{

float weight;

float height;

float bmi;

Scanner sc = new Scanner(System.in);

System.out.println("Enter height in centimeters: ");

height = sc.nextFloat();

height = height/100;

System.out.println("Enter weight in KGs :");

weight = sc.nextFloat();

bmi = weight / (height\*height);

System.out.println("The body mass index is : "+bmi+" kg/m2");

if(bmi < 18.5)

{

System.out.println("You are under-weight");

}

else if(bmi <= 24.9 && bmi >= 18.5)

{

System.out.println("You are in Healthy-weight");

}

else if(bmi <= 29.9 && bmi >= 25)

{

System.out.println("You are over-weight");

}

else if(bmi <= 34.9 && bmi >= 30)

{

System.out.println("You are obese");

}

else if(bmi <= 39.9 && bmi >= 35)

{

System.out.println("You are severely obese");

}

else if(bmi >= 40)

{

System.out.println("You are morbidity obese");

}

}

}

Output:

Enter height in centimeters:

170

Enter weight in KGs :

55

The body mass index is : 19.031141 kg/m2

You are in Healthy-weight

6). package com.bov.assignment1;

/\* Rectangle Area and Perimeter: Build a program that prompts the user to enter the length and width of a rectangle.

Calculate and display the area and perimeter of the rectangle

using variables to store the input and calculated values. \*/

import java.util.Scanner;

class RectangleAreaAndPerimeter

{

public static void main(String args[])

{

int length;

int width;

int perimeter;

int area;

Scanner sc = new Scanner(System.in);

System.out.println("Enter the length of the rectangle : ");

length = sc.nextInt();

System.out.println("Enter width of the rectangle : ");

width = sc.nextInt();

perimeter = 2 \* (length + width);

area = length \* width;

System.out.println("The area of the rectangle is "+area+" and the perimeter of the rectangle Is : "+perimeter);

}

}

Output:

Enter the length of the rectangle :

5

Enter width of the rectangle :

3

The area of the rectangle is 15 and the perimeter of the rectangle Is : 16

7). package com.bov.assignment1;

/\* Currency Conversion: Design a program that converts an amount of money from one currency to another.

Prompt the user to enter the amount and the exchange rate.

Use variables to store the input values and the converted amount. \*/

import java.util.Scanner;

public class CurrencyConversion{

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

double rupeeToDollar = 0.012;

double dollarToRupee = 83.333;

double rupeeToUAEDirham = 0.045;

double uaeDirhamToRupee = 22.222;

double rupeeToSwissFranc = 0.011;

double swissFrancToRupee = 90.9090;

double rupeeToJapaneseYen = 1.75;

double japaneseYenToRupee = 0.5714;

double rupeeToEuro = 0.011;

double euroToRupee = 90.9090;

double result;

int choice;

System.out.println("Enter your conversion choice from below: \n 1. Rupee to Dollar \n 2. Dollar to Rupee \n 3. Rupee to UAE Dirham \n 4. UAE Dirham to Rupee \n 5. Rupee To Swiss Franc \n 6. Swiss Franc To Rupee \n 7. Rupee to Japanese Yen \n 8. Japanese Yen to Rupee \n 9. Rupee to Euro \n 10. Euro to Rupee \n Enter your choice: ");

choice = sc.nextInt();

switch(choice)

{

case 1:

System.out.println("Enter Rupee value to convert to Dollar: ");

double rupeeVal1 = sc.nextDouble();

result = rupeeVal1 \* rupeeToDollar;

System.out.println("The currency value after conversion from rupee to dollar is : "+result);

break;

case 2:

System.out.println("Enter Dollar value to convert to Rupee: ");

double dollarVal = sc.nextDouble();

result = dollarVal \* dollarToRupee;

System.out.println("The currency value after conversion from Dollar to Rupee is : "+result);

break;

case 3:

System.out.println("Enter Rupee value to convert to UAE Dirham: ");

double rupeeVal = sc.nextDouble();

result = rupeeVal \* rupeeToUAEDirham;

System.out.println("The currency value after conversion from rupee to UAE Dirham is : "+result);

break;

case 4:

System.out.println("Enter UAE Dirham value to convert to Rupee: ");

double uaeDirhamVal = sc.nextDouble();

result = uaeDirhamVal \* uaeDirhamToRupee;

System.out.println("The currency value after conversion from UAE Dirham to Rupee is : "+result);

break;

case 5:

System.out.println("Enter Rupee value to convert to Swiss Franc: ");

double rupeeVal2 = sc.nextDouble();

result = rupeeVal2 \* rupeeToSwissFranc;

System.out.println("The currency value after conversion from Rupee to Swiss Franc is : "+result);

break;

case 6:

System.out.println("Enter Swiss Franc value to convert to Rupee: ");

double swissFrancVal = sc.nextDouble();

result = swissFrancVal \* swissFrancToRupee;

System.out.println("The currency value after conversion from Swiss Franc to Rupee is : "+result);

break;

case 7:

System.out.println("Enter Rupee value to convert to Japanese Yen: ");

double rupeeVal3 = sc.nextDouble();

result = rupeeVal3 \* rupeeToJapaneseYen;

System.out.println("The currency value after conversion from Rupee to Japanese Yen is : "+result);

break;

case 8:

System.out.println("Enter Japanese Yen value to convert to Rupee: ");

double japaneseYenVal = sc.nextDouble();

result = japaneseYenVal \* japaneseYenToRupee;

System.out.println("The currency value after conversion from Japanese Yen to Rupee is : "+result);

break;

case 9:

System.out.println("Enter Rupee value to convert to Euro: ");

double rupeeVal4 = sc.nextDouble();

result = rupeeVal4 \* rupeeToEuro;

System.out.println("The currency value after conversion from Rupee to Euro is : "+result);

break;

case 10:

System.out.println("Enter Euro value to convert to Rupee: ");

double euroVal = sc.nextDouble();

result = euroVal \* euroToRupee;

System.out.println("The currency value after conversion from Euro to Rupee is : "+result);

break;

default:

System.out.println("Incorrect answer.");

}

}

}

Output:

Enter your conversion choice from below:

1. Rupee to Dollar

2. Dollar to Rupee

3. Rupee to UAE Dirham

4. UAE Dirham to Rupee

5. Rupee To Swiss Franc

6. Swiss Franc To Rupee

7. Rupee to Japanese Yen

8. Japanese Yen to Rupee

9. Rupee to Euro

10. Euro to Rupee

Enter your choice:

10

Enter Euro value to convert to Rupee:

150

The currency value after conversion from Euro to Rupee is : 13636.35

8). package com.bov.assignment1;

/\* Time Conversion: Develop a program that converts time from seconds to hours, minutes, and seconds.

Prompt the user to enter the time in seconds, and use variables to store

the calculated values for hours, minutes, and remaining seconds. \*/

import java.util.Scanner;

class TimeConversion

{

public static void main(String args[])

{

int seconds;

int hours;

int minutes;

int remainingSeconds;

Scanner sc = new Scanner(System.in);

System.out.println("Enter time in seconds : ");

seconds = sc.nextInt();

remainingSeconds = seconds % 60;

minutes = (seconds % 3600) / 60;

hours = seconds / 3600;

System.out.println("The time is "+hours+" hours "+minutes+" minutes and "+remainingSeconds+" seconds");

}

}

Output:

Enter time in seconds :

49000

The time is 13 hours 36 minutes and 40 seconds

9). package com.bov.assignment1;

/\* Create a Class Hierarchy: Define a class hierarchy representing different types of vehicles.

Start with a base class "Vehicle" and create derived classes such as "Car," "Motorcycle," and "Truck."

Add relevant attributes and methods to each class that capture their unique characteristics. \*/

class Vehicle

{

void tyres(int a)

{

System.out.println("The vehicle has "+a+" tyres");

}

void doors(int b)

{

System.out.println("The vehicle has "+b+" doors");

}

void lights(int c)

{

System.out.println("The vehicle has "+c+" lights");

}

void fuel(String d)

{

System.out.println("The vehicle has "+d+" as its fuel");

}

}

class Car extends Vehicle

{

String color;

Car()

{

}

Car(String c)

{

this.color = c;

System.out.println("This car color is : "+color);

}

}

class MotorCycle extends Vehicle

{

String color;

MotorCycle()

{

}

MotorCycle(String c)

{

this.color = c;

System.out.println("This MotorCycle color is : "+color);

}

}

class Truck extends Vehicle

{

String color;

Truck()

{

}

Truck(String c)

{

this.color = c;

System.out.println("This Truck color is : "+color);

}

}

class Test

{

public static void main(String args[])

{

Car c = new Car("Violet");

System.out.println("The car specs are : ");

c.tyres(4);

c.doors(4);

c.lights(8);

c.fuel("Diesel");

MotorCycle m = new MotorCycle("Green");

System.out.println("The MotorCycle has following specs : ");

m.tyres(2);

m.doors(0);

m.lights(5);

m.fuel("Petrol");

Truck t = new Truck("Magenta");

System.out.println("The truck vehicle specs are : ");

t.tyres(10);

t.doors(2);

t.lights(8);

t.fuel("Petrol");

}

}

Output:

This car color is : Violet

The car specs are :

The vehicle has 4 tyres

The vehicle has 4 doors

The vehicle has 8 lights

The vehicle has Diesel as its fuel

This MotorCycle color is : Green

The MotorCycle has following specs :

The vehicle has 2 tyres

The vehicle has 0 doors

The vehicle has 5 lights

The vehicle has Petrol as its fuel

This Truck color is : Magenta

The truck vehicle specs are :

The vehicle has 10 tyres

The vehicle has 2 doors

The vehicle has 8 lights

The vehicle has Petrol as its fuel

10). package assignment1Part2;

/\* Implement Polymorphism: Extend the vehicle hierarchy assignment by implementing polymorphic behavior.

Create a method, such as "drive," in the base class, and override it in the derived classes to provide specific implementations.

Demonstrate polymorphic behavior by invoking the method on different objects of the derived classes. \*/

class Vehicle

{

void tyres(int a)

{

System.out.println("The vehicle has "+a+" tyres");

}

void doors(int b)

{

System.out.println("The vehicle has "+b+" doors");

}

void lights(int c)

{

System.out.println("The vehicle has "+c+" lights");

}

void fuel(String d)

{

System.out.println("The vehicle has "+d+" as its fuel");

}

void drive()

{

System.out.println("This is super class drive method");

}

}

class Car extends Vehicle

{

String color;

Car()

{

}

Car(String c)

{

this.color = c;

System.out.println("This car color is : "+color);

}

void drive()

{

super.drive();

System.out.println("Try to drive on left lane");

}

void drive(String a, String b)

{

System.out.println("Drive in "+a+" lane first and then"+b+" lane");

}

}

class MotorCycle extends Vehicle

{

String color;

MotorCycle()

{

}

MotorCycle(String c)

{

this.color = c;

System.out.println("This MotorCycle color is : "+color);

}

void drive()

{

super.drive();

System.out.println("Try to drive on left lane");

}

void drive(String a, String b)

{

System.out.println("Drive in "+a+" lane first and then"+b+" lane");

}

}

class Truck extends Vehicle

{

String color;

Truck()

{

}

Truck(String c)

{

this.color = c;

System.out.println("This Truck color is : "+color);

}

void drive()

{

super.drive();

System.out.println("Try to drive on left lane");

}

void drive(String a, String b)

{

System.out.println("Drive in "+a+" lane first and then"+b+" lane");

}

}

class Test

{

public static void main(String args[])

{

System.out.println("\n The car specs are : ");

Car c = new Car("Violet");

c.tyres(4);

c.doors(4);

c.lights(8);

c.fuel("Diesel");

c.drive();

c.drive("Right","Left");

System.out.println("\n The MotorCycle has following specs : ");

MotorCycle m = new MotorCycle("Green");

m.tyres(2);

m.doors(0);

m.lights(5);

m.fuel("Petrol");

m.drive();

m.drive("Left","Right");

System.out.println("\n The truck vehicle specs are : ");

Truck t = new Truck("Magenta");

t.tyres(10);

t.doors(2);

t.lights(8);

t.fuel("Petrol");

t.drive();

t.drive("Right", "Left");

}

}

Output:

The car specs are :

This car color is : Violet

The vehicle has 4 tyres

The vehicle has 4 doors

The vehicle has 8 lights

The vehicle has Diesel as its fuel

This is super class drive method

Try to drive on left lane

Drive in Right lane first and thenLeft lane

The MotorCycle has following specs :

This MotorCycle color is : Green

The vehicle has 2 tyres

The vehicle has 0 doors

The vehicle has 5 lights

The vehicle has Petrol as its fuel

This is super class drive method

Try to drive on left lane

Drive in Left lane first and thenRight lane

The truck vehicle specs are :

This Truck color is : Magenta

The vehicle has 10 tyres

The vehicle has 2 doors

The vehicle has 8 lights

The vehicle has Petrol as its fuel

This is super class drive method

Try to drive on left lane

Drive in Right lane first and thenLeft lane

11). package assignmentpart3;

/\* Shape Inheritance: Create a class hierarchy representing different shapes, such as "Circle," "Rectangle," and "Triangle."

Each shape should have common attributes like area and perimeter, as well as specific attributes and methods.

Implement inheritance and polymorphism to calculate and display the area and perimeter of different shapes. \*/

class Shape

{

void perimeterShow(double perimeter)

{

System.out.println("The perimeter is "+perimeter);

}

void areaShow(double area)

{

System.out.println("The Area is : "+area);

}

void sides()

{

System.out.println("This has 0 sides");

}

void sides(int a)

{

System.out.println("This has "+a+" sides");

}

}

class Circle extends Shape

{

int radius;

double perimeter;

double area;

Circle()

{

}

Circle(int radius)

{

this.radius = radius;

perimeter = 2\*3.14\*radius;

area = 3.14\*radius\*radius;

}

void sides()

{

super.sides();

System.out.println("Circle has 0 sides");

}

}

class Rectangle extends Shape{

int length;

int width;

float perimeter;

float area;

Rectangle()

{

}

Rectangle(int l, int w)

{

this.length = l;

this.width = w;

perimeter = 2\*(l+w);

area = l\*w;

}

}

class Triangle extends Shape

{

int side;

int height;

double perimeter;

double area;

Triangle()

{

}

Triangle(int side, int height)

{

perimeter = 3\*side;

area = 0.5\*side\*height;

}

}

class Test

{

public static void main(String args[])

{

Circle c = new Circle(3);

c.sides();

c.perimeterShow(c.perimeter);

c.areaShow(c.area);

Rectangle r = new Rectangle(4,5);

r.sides(4);

r.perimeterShow(r.perimeter);

r.areaShow(r.area);

Triangle t = new Triangle(2,4);

t.sides(3);

t.perimeterShow(t.perimeter);

t.areaShow(t.area);

}

}

Output:

This has 0 sides

Circle has 0 sides

The perimeter is 18.84

The Area is : 28.259999999999998

This has 4 sides

The perimeter is 18.0

The Area is : 20.0

This has 3 sides

The perimeter is 6.0

The Area is : 4.0

12). package assignmentpart3;

/\* Employee Inheritance: Design an employee management system using inheritance.

Create a base class "Employee" and derived classes such as "Manager," "Developer," and "Tester."

Each derived class should have specific attributes and methods, along with common attributes like name and salary.

Implement functionality to calculate the total salary of all employees. \*/

class Employee

{

private int id;

private String name;

private float salary;

void setId(int id)

{

this.id = id;

}

int getId()

{

return id;

}

void setName(String name)

{

this.name = name;

}

String getName()

{

return name;

}

void setSalary(float salary)

{

this.salary = salary;

}

float getSalary()

{

return salary;

}

}

class Manager extends Employee

{

int id;

String name;

float salary;

Manager()

{

}

Manager(int id, String name, float salary)

{

Manager m1 = new Manager();

m1.setId(id);

m1.setName(name);

m1.setSalary(salary);

}

}

class Developer extends Employee

{

int id;

String name;

float salary;

Developer()

{

}

Developer(int id, String name, float salary)

{

Developer m2 = new Developer();

m2.setId(id);

m2.setName(name);

m2.setSalary(salary);

}

}

class Tester extends Employee

{

int id;

String name;

float salary;

Tester()

{

}

Tester(int id, String name, float salary)

{

Tester m3 = new Tester();

m3.setId(id);

m3.setName(name);

m3.setSalary(salary);

}

}

class EmployeeTest

{

public static void main(String args[])

{

Manager m = new Manager(1,"Girish",65000);

System.out.println("Manager ID is : "+m.getId());

System.out.println("Manager Name is : "+m.getName());

System.out.println("Manager Salary is : "+m.getSalary());

Developer d = new Developer(2,"Rakesh",75000);

System.out.println("Developer ID is : "+d.getId());

System.out.println("Developer Name is : "+d.getName());

System.out.println("Developer Salary is : "+d.getSalary());

Tester t = new Tester(3,"Nitish",70000);

System.out.println("Manager ID is : "+t.getId());

System.out.println("Manager Name is : "+t.getName());

System.out.println("Manager Salary is : "+t.getSalary());

}

}

Output:

Manager ID is : 0

Manager Name is : null

Manager Salary is : 0.0

Developer ID is : 0

Developer Name is : null

Developer Salary is : 0.0

Manager ID is : 0

Manager Name is : null

Manager Salary is : 0.0

13). package assignmentpart3;

/\* Bank Account Inheritance: Create a class hierarchy representing different types of bank accounts,

such as "SavingsAccount" and "CurrentAccount." Implement common functionalities like deposit and withdrawal in the base class

and specific functionalities like interest calculation in the derived classes.

\*/

import java.util.Scanner;

import java.lang.System;

class BankAccount

{

Scanner sc = new Scanner(System.in);

float withdrawal(float cashWithdraw, float initialAmount, int pin)

{

float finalAmount = 0;

int count = 0;

int transaCompl = 0;

if(cashWithdraw <= initialAmount)

{

while(count < 3 && transaCompl == 0)

{

System.out.println("Enter pin for transaction to proceed : ");

int userPin = sc.nextInt();

if(userPin == pin)

{

System.out.println("Transaction Processing...");

finalAmount = initialAmount - cashWithdraw;

System.out.println("The balance in your account is : "+finalAmount);

transaCompl = 1;

}

else

{

System.out.println("wrong PIN");

count = count + 1;

}

}

}

else

{

System.out.println("Transaction declined due to Insufficient funds, sorry for the inconvenience...");

finalAmount = initialAmount;

}

return finalAmount;

}

float deposit(float cashDeposit, float initialAmount)

{

float finalAmount;

finalAmount = initialAmount + cashDeposit;

System.out.println("Amount "+cashDeposit+" deposited into your bank account");

System.out.println("Final Amount is"+finalAmount);

return finalAmount;

}

void showBalance(float accountBalance)

{

System.out.println("The account Balance is : "+accountBalance);

}

int setPin()

{

int pin;

System.out.println("Please enter pin that you want to set : ");

pin = sc.nextInt();

System.out.println("PIN set successfully...");

return pin;

}

}

class SavingsAccount extends BankAccount

{

private String accountHolderName = "K. VIJAY";

private String accountNumber = "09347838746";

private static int pin = 123456;

private static float initialAmount = 50000;

void calculateInterest(int time)

{

float interest = (initialAmount\*time\*2)/100;

System.out.println("The interest for your requested time frame "+time+" years is : "+interest+" and the total amount will be : "+(initialAmount+interest));

}

String getAccountHolderName()

{

return accountHolderName;

}

String getAccountNumber()

{

return accountNumber;

}

void setPIN()

{

pin = setPin();

}

void showBalance()

{

super.showBalance(initialAmount);

}

void deposit(float depositAmount)

{

initialAmount = super.deposit(depositAmount, initialAmount);

}

void withdrawal(float cashWithdraw)

{

initialAmount = super.withdrawal(cashWithdraw, initialAmount, pin);

}

}

class CurrentAccount extends BankAccount

{

private String accountHolderName = "G. VINAY";

private String accountNumber = "09234567891";

private static int pin = 234567;

private static float initialAmount = 100000;

void calculateInterest(int time)

{

float interest = (initialAmount\*time\*5)/100;

System.out.println("The interest for your requested time frame "+time+" years is : "+interest+" and the total amount will be : "+(initialAmount+interest));

}

String getAccountHolderName()

{

return accountHolderName;

}

String getAccountNumber()

{

return accountNumber;

}

float getInitialAmount()

{

return initialAmount;

}

void setPIN()

{

pin = setPin();

}

void showBalance()

{

super.showBalance(initialAmount);

}

void deposit(float depositAmount)

{

initialAmount = super.deposit(depositAmount, initialAmount);

}

void withdrawal(float cashWithdraw)

{

initialAmount = super.withdrawal(cashWithdraw, initialAmount, pin);

}

}

class BankAccountTest

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

int choiceAccount;

int choice;

System.out.println("Enter an option \n 1. Savings Account \n 2. Current Account : ");

choiceAccount = sc.nextInt();

switch(choiceAccount)

{

case 1:

SavingsAccount sa = new SavingsAccount();

System.out.println("Your Savings account details are as follows : ");

System.out.println("Account Holder Name : "+sa.getAccountHolderName());

System.out.println("Account Number :"+sa.getAccountNumber());

System.out.println("Choose from below options : \n 1. Show Balance \n 2. Deposit \n 3. Withdrawal \n 4. exit() \n Enter : ");

choice = sc.nextInt();

switch(choice)

{

case 1:

sa.showBalance();

break;

case 2:

System.out.println("Enter the amount you want to deposit : ");

float depositAmount = sc.nextFloat();

sa.deposit(depositAmount);

break;

case 3:

System.out.println("Enter the amount you want to withdraw : ");

float withdrawalAmount = sc.nextFloat();

sa.withdrawal(withdrawalAmount);

break;

case 4:

exit();

break;

default:

System.out.println("Incorrect answer...");

}

break;

case 2:

CurrentAccount ca = new CurrentAccount();

System.out.println("Your Current account details are as follows : ");

System.out.println("Account Holder Name : "+ca.getAccountHolderName());

System.out.println("Account Number :"+ca.getAccountNumber());

System.out.println("Choose from below options : \n 1. Show Balance \n 2. Deposit \n 3. Withdrawal \n 4. exit() \n Enter : ");

choice = sc.nextInt();

switch(choice)

{

case 1:

ca.showBalance();

break;

case 2:

System.out.println("Enter the amount you want to deposit : ");

float depositAmount = sc.nextFloat();

ca.deposit(depositAmount);

break;

case 3:

System.out.println("Enter the amount you want to withdraw : ");

float withdrawalAmount = sc.nextFloat();

ca.withdrawal(withdrawalAmount);

break;

case 4:

exit();

break;

default:

System.out.println("Incorrect answer, please try again...");

}

break;

default:

System.out.println("Incorrect Answer");

}

}

private static int exit() {

// TODO Auto-generated method stub

System.out.println("You are exited...");

return 0;

}

}

Output:

Enter an option

1. Savings Account

2. Current Account :

1

Your Savings account details are as follows :

Account Holder Name : K. VIJAY

Account Number :09347838746

Choose from below options :

1. Show Balance

2. Deposit

3. Withdrawal

4. exit()

Enter :

2

Enter the amount you want to deposit :

15334

Amount 15334.0 deposited into your bank account

Final Amount is65334.0

14). package assignmentpart3;

/\* Animal Inheritance: Design an animal hierarchy with a base class "Animal" and derived classes such as "Cat," "Dog," and "Bird."

Add appropriate methods and attributes to represent different behaviors and characteristics of each animal.

Implement polymorphism by invoking common methods on objects of different animal types.

\*/

class Animal

{

void speed(int flyingSpeed)

{

System.out.println("The flying speed of the bird is "+flyingSpeed+" km/h");

}

void speed(double minSpeed, int maxSpeed)

{

System.out.println("The minimum speed is "+minSpeed+" m/s and maximum speed is "+maxSpeed+" m/s");

}

void color(String animalColor)

{

System.out.println("The color of the animal is : "+animalColor);

}

void height(int height)

{

System.out.println("The height of the animal is : "+height+" cm");

}

}

class Cat extends Animal

{

String name = "Cat";

int height;

double minS;

int maxS;

String color;

Cat(int h, double min, int max, String color)

{

this.height = h;

this.minS = min;

this.maxS = max;

this.color = color;

super.height(height);

super.color(color);

super.speed(minS, maxS);

}

}

class Dog extends Animal

{

String name = "Dog";

int height;

double minS;

int maxS;

String color;

Dog(int h, double min, int max, String color)

{

this.height = h;

this.minS = min;

this.maxS = max;

this.color = color;

height(height);

color(color);

}

void speed()

{

System.out.println("The min speed of the animal is : "+minS+" m/s and max speed is : "+ maxS+" m/s");

}

}

class Bird extends Animal

{

String name = "Bird";

int height;

int speed;

String color;

Bird(int h, int speed, String color)

{

this.height = h;

this.speed = speed;

this.color = color;

height(height);

color(color);

speed(speed);

}

}

class AnimalTest

{

public static void main(String args[])

{

Cat c = new Cat(10, 0.1, 15, "White-Brown");

System.out.println("Animal name : "+c.name);

Dog d = new Dog(15, 0.2, 10, "Brown");

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

d.speed();

Bird b = new Bird(5, 12, "White");

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

}

}

Output:

The height of the animal is : 10 cm

The color of the animal is : White-Brown

The minimum speed is 0.1 m/s and maximum speed is 15 m/s

Animal name : Cat

The height of the animal is : 15 cm

The color of the animal is : Brown

Animal Name : Dog

The min speed of the animal is : 0.2 m/s and max speed is : 10 m/s

The height of the animal is : 5 cm

The color of the animal is : White

The flying speed of the bird is 12 km/h

Animal Name : Bird

15). package assignmentpart3;

/\*

School System Inheritance: Model a school system using inheritance.

Create a base class "Person" and derived classes such as "Teacher" and "Student."

Implement functionalities like assigning subjects to teachers and registering students for courses.

Demonstrate inheritance and polymorphism by accessing and displaying information about different types of people in the school system.

\*/

import java.util.Scanner;

class Person

{

private String[] students={"Anil", "Sunil", "Raju", "Pratish", "Sunitha", "Geetha", "Ganesh"};

private int[] roll = {13,12,11,10,9,8,7};

private int[] classname = {1,4,2,3,5,7,8};

protected int[] courseNum = new int[7];

protected String course[] = new String[7];

private int[] teacherId = {1,2,3};

private String[] teachers = {"Rambabu", "Pratap", "Pranitha"};

private String[] qualification = {"PhD", "PhD", "PhD"};

protected String[] subjects = new String[3];

public void showStudent(int rollno)

{

int found = 0;

for(int i=0; i<7; i++)

{

if(roll[i] == rollno)

{

found = 1;

System.out.println("Student with roll no.: "+rollno+" is : "+students[i]+" , class name : "+classname[i]+", course number : "+courseNum[i]+", and course is : "+course[i]);

}

}

if(found == 0)

{

System.out.println("The student with roll no.: "+rollno+" is not present");

}

}

public void showTeacher(int id)

{

int found = 0;

for(int i=0; i<3; i++)

{

if(teacherId[i] == id)

{

found = 1;

System.out.println("Teacher with Teacher ID: "+id+" is : "+teachers[i]+" and qualification is : "+qualification[i]+" and assigned sunject is : "+subjects[i]);

}

}

if(found == 0)

{

System.out.println("Teacher with ID : "+id+" is not present.");

}

}

int studentFound(String n, int r)

{

for(int i=0; i<7; i++)

{

if(students[i] == n && roll[i] == r)

{

return i;

}

}

return -1;

}

public void getTeacherInfo()

{

for(int i = 0; i < 3; i++)

{

System.out.println("Teacher : "+teachers[i]+" \n Qualification : "+qualification[i]);

}

}

public void getTeacherInfo(int a)

{

System.out.println("Teacher : "+teachers[a]+" \n Qualification : "+qualification[a]);

}

public void getStudentInfo(int a)

{

System.out.println("Student : "+students[a]+" \n Roll No. : "+roll[a]+" \n Class Name : "+classname[a]);

}

int teacherFound(String t, String q)

{

for(int i=0; i<3; i++)

{

if(teachers[i] == t && qualification[i] == q)

{

return i;

}

}

return -1;

}

public void getStudentInfo()

{

for(int i = 0; i < 7; i++)

{

System.out.println("Student : "+students[i]+" \n Roll No. : "+roll[i]+" \n Class Name : "+classname[i]);

}

}

public void studentStatus(boolean a, int c, String name, int p)

{

if(a)

{

courseNum[p] = c;

course[p] = name;

System.out.println("Student registration success for course number : "+c+" , course name : "+name);

}

else

{

System.out.println("Student registration failed !!!");

}

}

public void teacherStatus(boolean a, String t, String subject, int pos)

{

if(a)

{

subjects[pos] = subject;

System.out.println("The teacher "+t+" is assigned "+subject);

}

else

{

System.out.println("Assignment of status failed !!!");

}

}

}

class Student extends Person

{

public void showStudent(int rollno)

{

super.showStudent(rollno);

}

public void getStudentInfo(int a)

{

super.getStudentInfo(a-1);

}

void register(int courseNum, String courseName, String studentName, int rollNo)

{

int pos;

if((pos = studentFound(studentName, rollNo)) != -1)

{

studentStatus(true, courseNum, courseName, pos);

}

else

{

studentStatus(false, courseNum, courseName, pos);

}

}

}

class Teacher extends Person

{

String subject;

public void showTeacher(int id)

{

super.showTeacher(id);

}

public void getTeacherInfo(int a)

{

super.getTeacherInfo(a-1);

}

void assign(String subjectName, String teacherName, String qualification)

{

int pos;

if((pos = teacherFound(teacherName, qualification)) != -1)

{

teacherStatus(true, teacherName, subjectName, pos);

}

else

{

teacherStatus(false, teacherName, subjectName, pos);

}

}

}

class PersonTest

{

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

int choice;

Student s = new Student();

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

s.getStudentInfo(1);

s.register(1, "B Tech", "Anil", 13);

s.register(2, "Degree", "Sunil", 12);

System.out.println("Do you want to display all students info type \"1\" if yes or \"2\" for individual students info : ");

choice = sc.nextInt();

if(choice == 1)

{

s.getStudentInfo();

}

else if(choice == 2)

{

int stroll;

System.out.println("Enter student roll no.: ");

stroll = sc.nextInt();

s.showStudent(stroll);

}

else

{

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

}

Teacher t = new Teacher();

System.out.println("Teacher at popsition 1 : ");

t.getTeacherInfo(1);

t.assign("Science", "Rambabu", "PhD");

t.assign("Math", "Pratap", "PhD");

System.out.println("Do you want to display all teachers info type \"1\" if yes and \"2\" if you want to access individual teacher info: ");

choice = sc.nextInt();

if(choice == 1)

{

s.getTeacherInfo();

}

else if(choice == 2)

{

int teid;

System.out.println("Enter teacher ID no.: ");

teid = sc.nextInt();

s.showTeacher(teid);

}

else

{

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

}

}

}

Output:

Student at position 1 :

Student : Anil

Roll No. : 13

Class Name : 1

Student registration success for course number : 1 , course name : B Tech

Student registration success for course number : 2 , course name : Degree

Do you want to display all students info type "1" if yes or "2" for individual students info :

1

Student : Anil

Roll No. : 13

Class Name : 1

Student : Sunil

Roll No. : 12

Class Name : 4

Student : Raju

Roll No. : 11

Class Name : 2

Student : Pratish

Roll No. : 10

Class Name : 3

Student : Sunitha

Roll No. : 9

Class Name : 5

Student : Geetha

Roll No. : 8

Class Name : 7

Student : Ganesh

Roll No. : 7

Class Name : 8

Teacher at popsition 1 :

Teacher : Rambabu

Qualification : PhD

The teacher Rambabu is assigned Science

The teacher Pratap is assigned Math

Do you want to display all teachers info type "1" if yes and "2" if you want to access individual teacher info:

1

Teacher : Rambabu

Qualification : PhD

Teacher : Pratap

Qualification : PhD

Teacher : Pranitha

Qualification : PhD