**ASSIGNMENT-12**

**Question-1**

**Problem Statement:** Create a class Vehicle as follows :

Data members (All are private) –[Brand, Country\_of\_Origin, Base\_price]

Methods - input (to input details of vehicle) and display (to show vehicle details).

Create a sub class Car as follows:

Data members – [Model, speed, Market\_price]

[NB: Market price of a car can be calculated from the Base price and speed. If speed is above 80km/hr, market price will be 15% more than the base price otherwise market price will be 5% less than the base price.]

Methods – read (to input car details) and show (to show car details).

In addition to above methods add more appropriate methods to set the required data members.

Now create objects. Input required data and show the details (Brand, Country\_of\_Origin, Base\_price, Model, speed, Market\_price) of any car.

**Source Code**

// Class Vehicle

import java.util.Scanner;

public class Vehicle {

private String brand;

private String country\_of\_origin;

private double base\_price;

// Constructor

public Vehicle() {

brand = "";

country\_of\_origin = "";

base\_price = 0.0;

}

// Getter

public double getBase\_price() {

return base\_price;

}

// Method to input details of vehicle

public void input() {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter brand:");

brand = scanner.nextLine();

System.out.println("Enter country of origin:");

country\_of\_origin = scanner.nextLine();

System.out.println("Enter base price:");

base\_price = scanner.nextDouble();

}

// Method to display vehicle details

public void display() {

System.out.println("Brand: " + brand);

System.out.println("Country of Origin: " + country\_of\_origin);

System.out.println("Base Price: $" + base\_price);

}

}

// SubClass Car

import java.util.Scanner;

class Car extends Vehicle {

private String model;

private double speed;

private double market\_price;

// Constructor

public Car() {

super();

model = "";

speed = 0.0;

market\_price = 0.0;

}

// Method to calculate market price

private void calculateMarketPrice() {

if (speed > 80) {

market\_price = getBase\_price() \* 1.15; // 15% more than base price

} else {

market\_price = getBase\_price() \* 0.95; // 5% less than base price

}

}

// Method to input car details

public void read() {

super.input();

Scanner scanner = new Scanner(System.in);

System.out.println("Enter model:");

model = scanner.nextLine();

System.out.println("Enter speed (in km/hr):");

speed = scanner.nextDouble();

calculateMarketPrice();

}

// Method to show car details

public void show() {

super.display();

System.out.println("Model: " + model);

System.out.println("Speed: " + speed + " km/hr");

System.out.println("Market Price: $" + market\_price);

}

}

//Class TestVehicle (with main method)

public class TestVehicle {

public static void main(String[] args) {

Car car = new Car();

System.out.println("Enter car details:");

car.read();

System.out.println("\nCar Details:");

car.show();

}

}

**OUTPUT:**

Enter car details:

Enter brand:

Mahindra

Enter country of origin:

India

Enter base price:

1400000

Enter model:

XUV 700

Enter speed (in km/hr):

220

Car Details:

Brand: Mahindra

Country of Origin: India

Base Price: $1400000.0

Model: XUV 700

Speed: 220.0 km/hr

Market Price: $1609999.99

**Question-2**

**Problem Statement:** Create a class Number:

Data member: An array of type integer.

Constructor: Constructor with one parameter n, that is the size of the array. Allocate n memory for the array and input n numbers into the array.

Method-1: To display all the values in the array.

Derive a class OddNum from the class Number:

Data member: An array of type integer.

Constructor: To count the odd numbers present in the array of its base class Number and accordingly allocate memory for its own array.

Method-1: To copy all odd numbers from its base class array to its own array.

Method-2: To display all odd numbers.

Derive a class PrimeNum from the class OddNum:

Data member: An array of type integer.

Constructor: To count the prime numbers present in the array of its base class OddNum and accordingly allocate memory for its own array.

Method-1: To copy all prime numbers from its base class array to its own array.

Method-2: To display all prime numbers.

**Source Code**

// Class Number

import java.util.Scanner;

public class Number {

protected int[] array;

// Constructor to initialize the array and input numbers

public Number(int n) {

array = new int[n];

Scanner scanner = new Scanner(System.in);

System.out.println("Enter " + n + " numbers:");

for (int i = 0; i < n; i++) {

array[i] = scanner.nextInt();

}

}

// Method to display all values in the array

public void display() {

System.out.println("Array Values:");

for (int num : array) {

System.out.print(num + " ");

}

System.out.println();

}

}

//Class OddNum

public class OddNum extends Number {

protected int[] oddArray;

protected int oddCount;

// Constructor to count odd numbers and allocate memory

public OddNum(int n) {

super(n);

oddCount = 0;

for (int num : array) {

if (num % 2 != 0) {

oddCount++;

}

}

oddArray = new int[oddCount];

copyOddNumbers();

}

// Method to copy odd numbers to own array

private void copyOddNumbers() {

int index = 0;

for (int num : array) {

if (num % 2 != 0) {

oddArray[index++] = num;

}

}

}

// Method to display all odd numbers

public void displayOddNumbers() {

System.out.println("Odd Numbers:");

for (int num : oddArray) {

System.out.print(num + " ");

}

System.out.println();

}

}

//Class PrimeNum

public class PrimeNum extends OddNum {

private int[] primeArray;

private int primeCount;

// Constructor to count prime numbers and allocate memory

public PrimeNum(int n) {

super(n);

primeCount = 0;

for (int num : oddArray) {

if (isPrime(num)) {

primeCount++;

}

}

primeArray = new int[primeCount];

copyPrimeNumbers();

}

// Method to check if a number is prime

private boolean isPrime(int num) {

if (num <= 1) {

return false;

}

for (int i = 2; i <= Math.sqrt(num); i++) {

if (num % i == 0) {

return false;

}

}

return true;

}

// Method to copy prime numbers to own array

private void copyPrimeNumbers() {

int index = 0;

for (int num : oddArray) {

if (isPrime(num)) {

primeArray[index++] = num;

}

}

}

// Method to display all prime numbers

public void displayPrimeNumbers() {

System.out.println("Prime Numbers:");

for (int num : primeArray) {

System.out.print(num + " ");

}

System.out.println();

}

}

// Class TestNum

import java.util.Scanner;

public class TestNum {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int size = scanner.nextInt();

PrimeNum primeNum = new PrimeNum(size);

System.out.println("\nAll Numbers:");

primeNum.display();

System.out.println("\nOdd Numbers:");

primeNum.displayOddNumbers();

System.out.println("\nPrime Numbers:");

primeNum.displayPrimeNumbers();

}

}

**OUTPUT:**

Enter the size of the array:

8

Enter 8 numbers:

77

36

45

29

91

69

13

24

All Numbers:

Array Values:

77 36 45 29 91 69 13 24

Odd Numbers:

Odd Numbers:

77 45 29 91 69 13

Prime Numbers:

Prime Numbers:

29 13