**OOPM Lab**

**Lab Assingment number 06**

**Name:** Aamir Ansari **Batch:** A **Roll no.** 01

**Aim:** Write a JAVA program to implement the concept of Interface

**Problem statement:**

Create an interface vehicle and classes like Bicycle, Bike and Car having common functionalities like and put all the common functionalities in the interface. Classes like Bicycle, Bike and Car implement all these functionalities in their own class in their own way. Write another class to use these functionalities from at least one class.

**Theory**:

1. Interface

An interface is a reference type in Java. It is similar to class. It is a collection of abstract methods. A class implements an interface, thereby inheriting the abstract methods of the interface.

Along with abstract methods, an interface may also contain constants, default methods, static methods, and nested types. Method bodies exist only for default methods and static methods.

Writing an interface is similar to writing a class. But a class describes the attributes and behaviors of an object. And an interface contains behaviors that a class implements.

1. Why and When to use

In JAVA multiple inheritance is not primitively allowed, in order to perform multiple inheritance, Interface is used.

Interface is used to achieve absolute abstraction.

Interface is used to achieve loose coupling.

1. Interface with Inheritance example

// example

public class MammalInt implements Animal {

public void eat() {

System.out.println("Mammal eats");

}

public void travel() {

System.out.println("Mammal travels");

}

public int noOfLegs() {

return 0;

}

public static void main(String args[]) {

MammalInt m = new MammalInt();

m.eat();

m.travel();

}

}

// output

Mammal eats

Mammal travels

**Code (A)**

import java.util.\*;

interface Vehicle {

void setDescriptions();

void displayDescription();

}

class Bicycle implements Vehicle {

String companyName = "";

String color = "";

int registrationNumber = 0;

int numberOfWheels = 0;

// gets and sets information of Bicycle

public void setDescriptions() {

Scanner sc = new Scanner(System.in);

System.out.print("Enter name of the company : ");

this.companyName = sc.nextLine();

System.out.print("Enter colour of the Bicycle : ");

this.color = sc.nextLine();

System.out.print("Enter registration number of the Bicycle : ");

this.registrationNumber = sc.nextInt();

System.out.print("Enter number of wheels of the Bicycle : ");

this.numberOfWheels = sc.nextInt();

}

// display information of Bicycle

public void displayDescription() {

System.out.println();

System.out.println("\*\*\* Description of BICYCLE \*\*\*");

System.out.println("COMPANY : " + this.companyName);

System.out.println("COLOUR : " + this.color);

System.out.println("Reg Number : " + this.registrationNumber);

System.out.println("No of Wheels : " + this.numberOfWheels);

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println();

}

}

class Bike implements Vehicle {

String companyName = "";

String color = "";

int registrationNumber = 0;

String namePlate = "";

int numberOfWheels = 0;

// gets and sets information of Bike

public void setDescriptions() {

Scanner sc = new Scanner(System.in);

System.out.print("Enter name of the company : ");

this.companyName = sc.nextLine();

System.out.print("Enter colour of the Bike : ");

this.color = sc.nextLine();

System.out.print("Enter registration number of the Bike : ");

this.registrationNumber = sc.nextInt();

sc.nextLine();

System.out.print("Enter name plate of the Bike : ");

this.namePlate = sc.nextLine();

System.out.print("Enter number of wheels of the Bike : ");

this.numberOfWheels = sc.nextInt();

}

// display information of Bike

public void displayDescription() {

System.out.println();

System.out.println("\*\*\* Description of Bike \*\*\*");

System.out.println("Company : " + this.companyName);

System.out.println("Colour : " + this.color);

System.out.println("Reg Number : " + this.registrationNumber);

System.out.println("Name plate : " + this.namePlate);

System.out.println("No of Wheels : " + this.numberOfWheels);

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println();

}

}

class Car implements Vehicle {

String companyName = "";

String color = "";

int registrationNumber = 0;

String namePlate = "";

int numberOfWheels = 0;

// gets and sets information of Car

public void setDescriptions() {

Scanner sc = new Scanner(System.in);

System.out.print("Enter name of the company : ");

this.companyName = sc.nextLine();

System.out.print("Enter colour of the Car : ");

this.color = sc.nextLine();

System.out.print("Enter registration number of the Car : ");

this.registrationNumber = sc.nextInt();

sc.nextLine();

System.out.print("Enter name plate of the Car : ");

this.namePlate = sc.nextLine();

System.out.print("Enter number of wheels of the Car : ");

this.numberOfWheels = sc.nextInt();

}

// display information of Car

public void displayDescription() {

System.out.println();

System.out.println("\*\*\* Description of CAR \*\*\*");

System.out.println("Company : " + this.companyName);

System.out.println("Colour : " + this.color);

System.out.println("Reg Number : " + this.registrationNumber);

System.out.println("Name plate : " + this.namePlate);

System.out.println("No of Wheels : " + this.numberOfWheels);

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println();

}

}

class InterfaceProgram {

public static void main(String args[]) {

Scanner sc = new Scanner(System.in);

int choice;

while (true) {

System.out.println("(1) Bicycle ");

System.out.println("(2) BIKE");

System.out.println("(3) CAR");

System.out.println("(4) EXIT");

System.out.print("Enter your choice : ");

choice = sc.nextInt();

switch (choice) {

case 1:

Bicycle objOfBicycle = new Bicycle();

objOfBicycle.setDescriptions();

objOfBicycle.displayDescription();

break;

case 2:

Bike objOfBike = new Bike();

objOfBike.setDescriptions();

objOfBike.displayDescription();

break;

case 3:

Car objOfCar = new Car();

objOfCar.setDescriptions();

objOfCar.displayDescription();

break;

case 4:

System.out.println("\*\*\* E X I T I N G \*\*\*");

System.exit(0);

default:

System.out.println("\*\*\* INVALID INPUT \*\*\*");

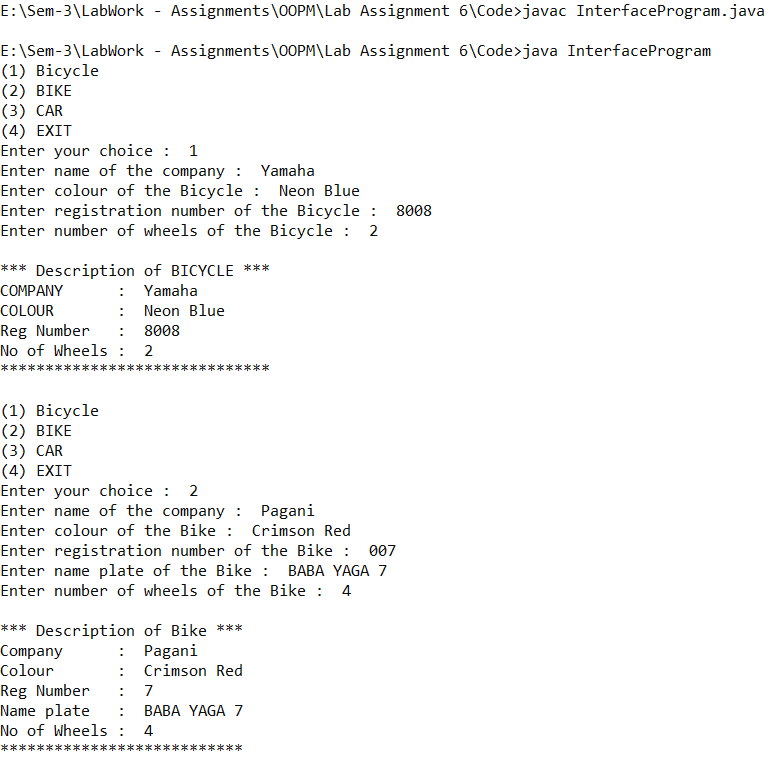
}

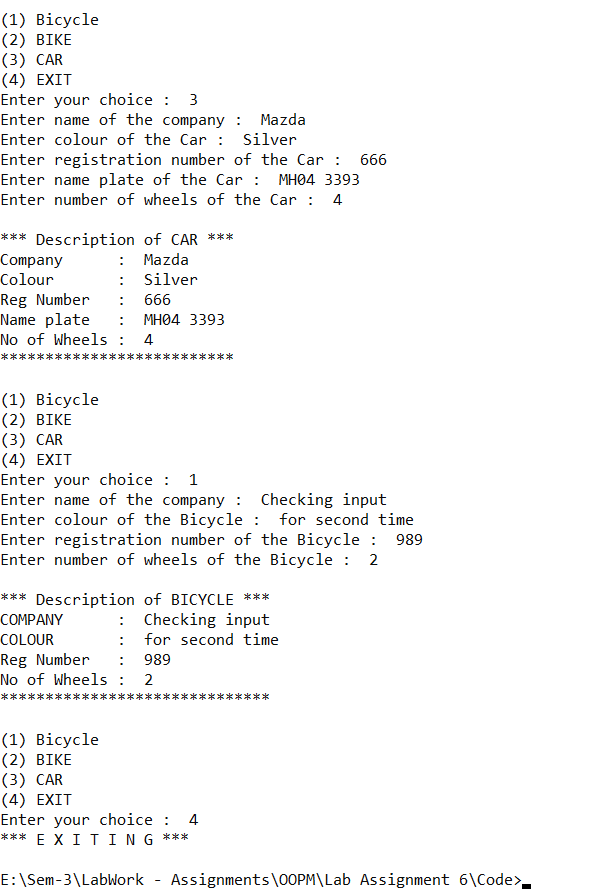
}

}

}

// output

.

.

**Code (B)**

import java.util.\*;

import java.lang.\*;

interface Dad {

void getXorY();

}

interface Mom {

void getX();

}

class Child implements Dad, Mom {

String XorY = "";

String X = "";

// get a gene from Dad

public void getXorY() {

if (Math.random() < 0.5) {

this.XorY = "Y";

} else {

this.XorY = "X";

}

}

// get a gene from Mom

public void getX() {

this.X = "X";

}

// Reveal the gender of child

public void genderReveal() {

String gender = this.X + this.XorY;

if (gender.equals("XX")) {

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

System.out.println("!! It's a GIRL !!");

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

} else if (gender.equals("XY"){

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

System.out.println("!! It's a BOY !!");

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

}

}

public static void main(String args[]) {

Child baby = new Child();

baby.getXorY();

baby.getX();

baby.genderReveal();

}

}

**//** output

