UNIT 2

ASSIGMENT 3

Name:- Sciddhanto Sinha
Roll No:- 2213111
Division – SY2

Assignment Title: Implement a concept of inheritance in Java.

Aim: Write a java program to create class car, truck and motorcycle which extends the vehicle class (attribute registration number, color, type of vehicle) with their own attribute like make, CC and fuel type. Input data from the user and print all the details.

Pre-Requisites: C/C++

Objective: Demonstrate and apply the concept of inheritance.

Outcomes: Students are able to Understand and Apply inheritance concepts with java programming

Theory:

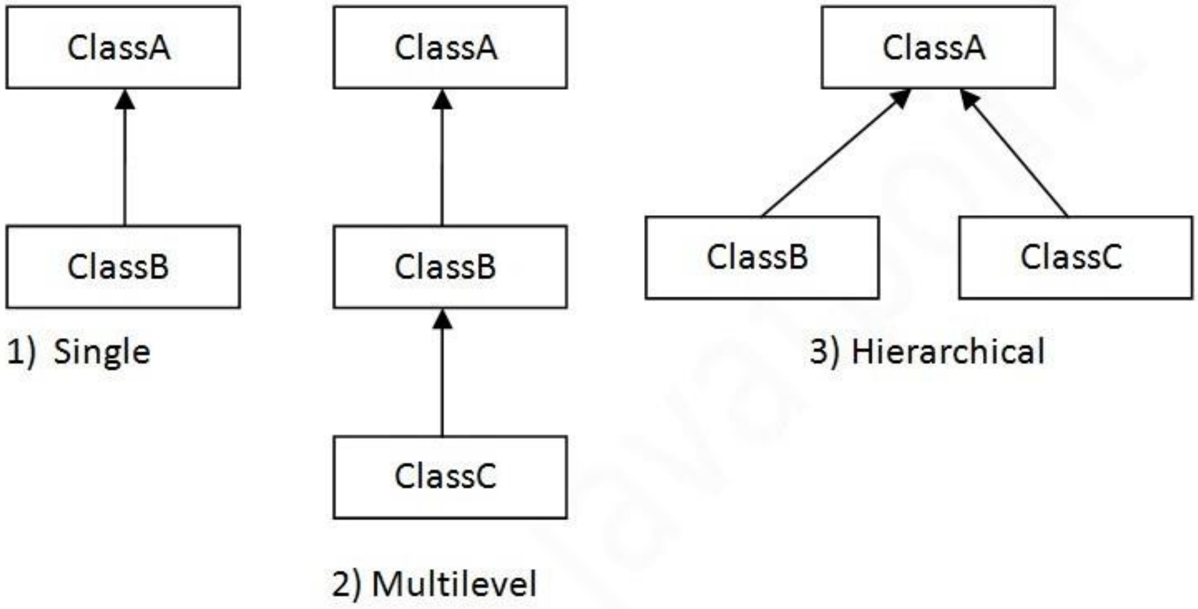
Inheritance in Java
Inheritance is an important pillar of OOP(Object-Oriented Programming). It is the mechanism in java by which one class is allowed to inherit the features(fields and methods) of another class. In Java, inheritance means creating new classes based on existing ones. A class that inherits from another class can reuse the methods and fields of that class. In addition, you can add new fields and methods to your current class as well.

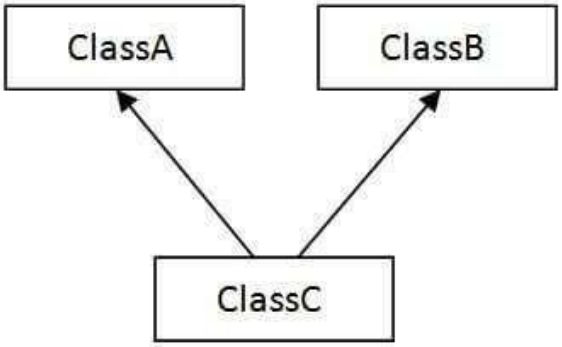
Important terminologies used in Inheritance:

- Class: Class is a set of objects which shares common characteristics/ behavior and common properties/ attributes. Class is not a real-world entity. It is just a template or blueprint or prototype from which objects are created.
- Super Class/Parent Class: The class whose features are inherited is known as a superclass(or a base class or a parent class).
- Sub Class/Child Class: The class that inherits the other class is known as a subclass(or a derived class, extended class, or child class). The subclass can add its own fields and methods in addition to the superclass fields and methods.
- Reusability: Inheritance supports the concept of “reusability”, i.e. when we want to create a new class and there is already a class that includes some of the code that

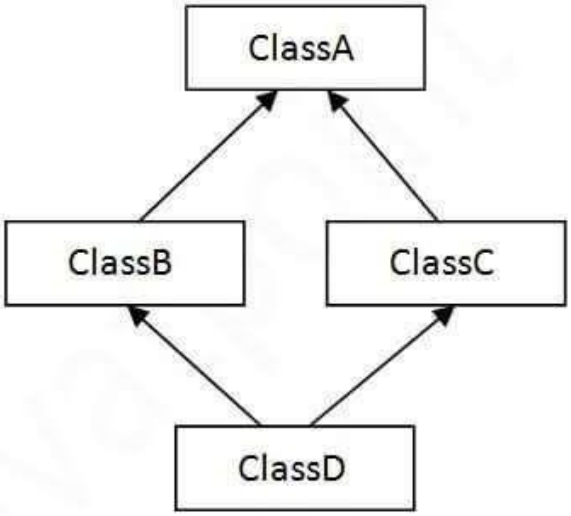
we want, we can derive our new class from the existing class. By doing this, we are reusing the fields and methods of the existing class.

Types of Inheritance:-





4) Multiple



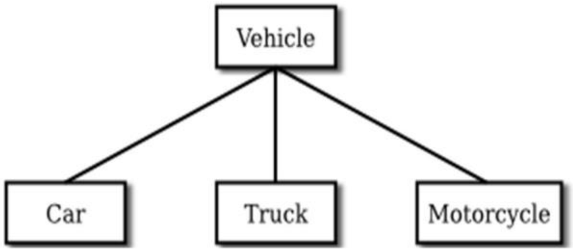
5) Hybrid

Example:-

```
class Animal{
void eat(){System.out.println("eating...");}
}
class Dog extends Animal{
void bark(){System.out.println("barking...");}
}
class TestInheritance{
public static void main(String args[]){
Dog d=new Dog();
d.bark();
d.eat();
}}
```

Algorithm/Steps:

- Define superclass vehicle
- Suppose that a program has to deal with motor vehicles, including cars, trucks, and motorcycles.
- The program could use a class named Vehicle to represent all types of vehicles.
- Since cars, trucks, and motorcycles are types of vehicles, they would be represented by subclasses of the Vehicle class, as shown in this class hierarchy diagram:



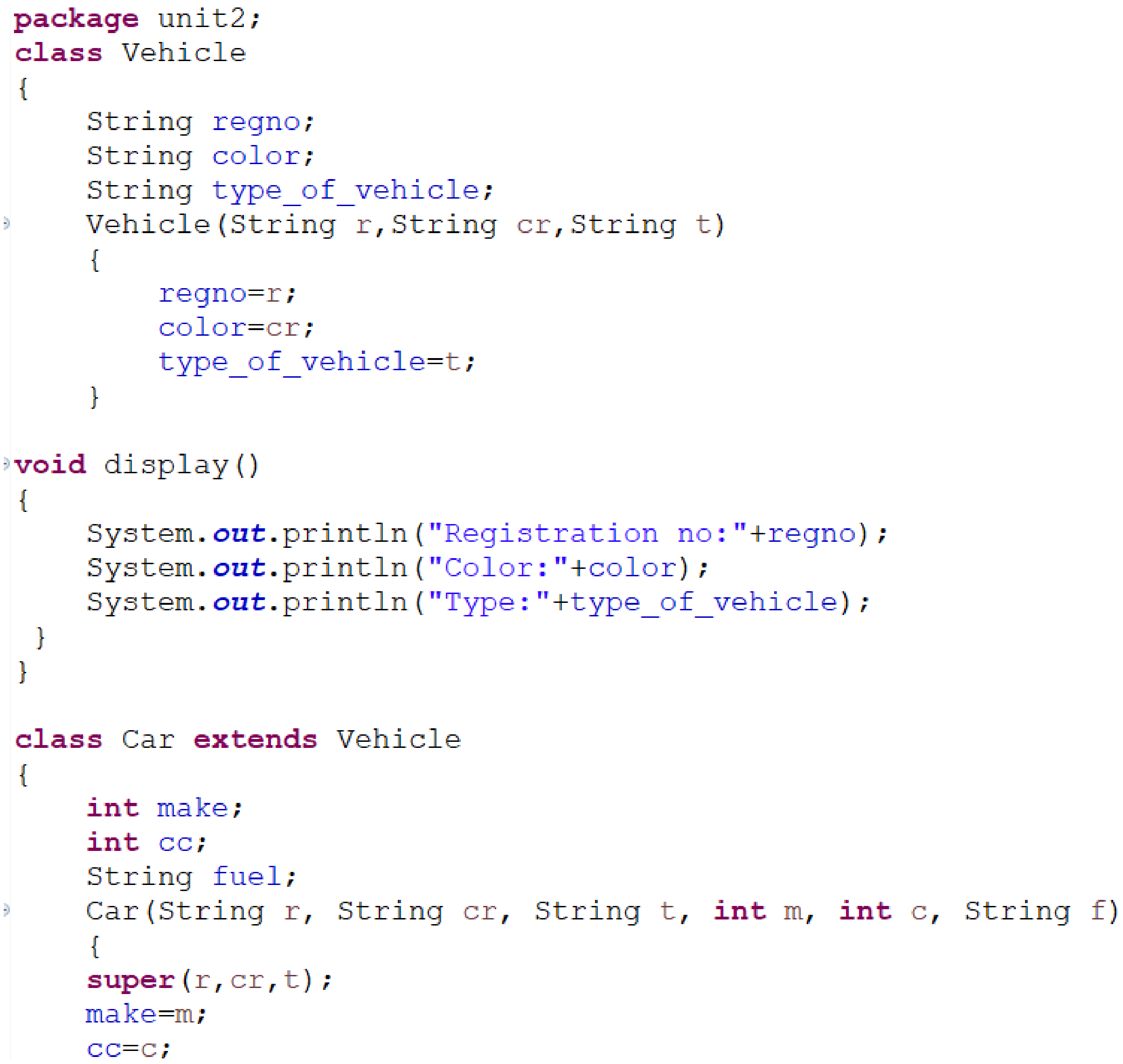
- The Vehicle class would include instance variables such as registration Number and owner and instance methods such as transferOwnership().
- These are variables and methods common to all vehicles.
- The three subclasses of Vehicle—Car, Truck, and Motorcycle—could then be used to hold variables and methods specific to particular types of vehicles

Conclusion: Hence we studied and implemented the concept of inheritance with java programming.

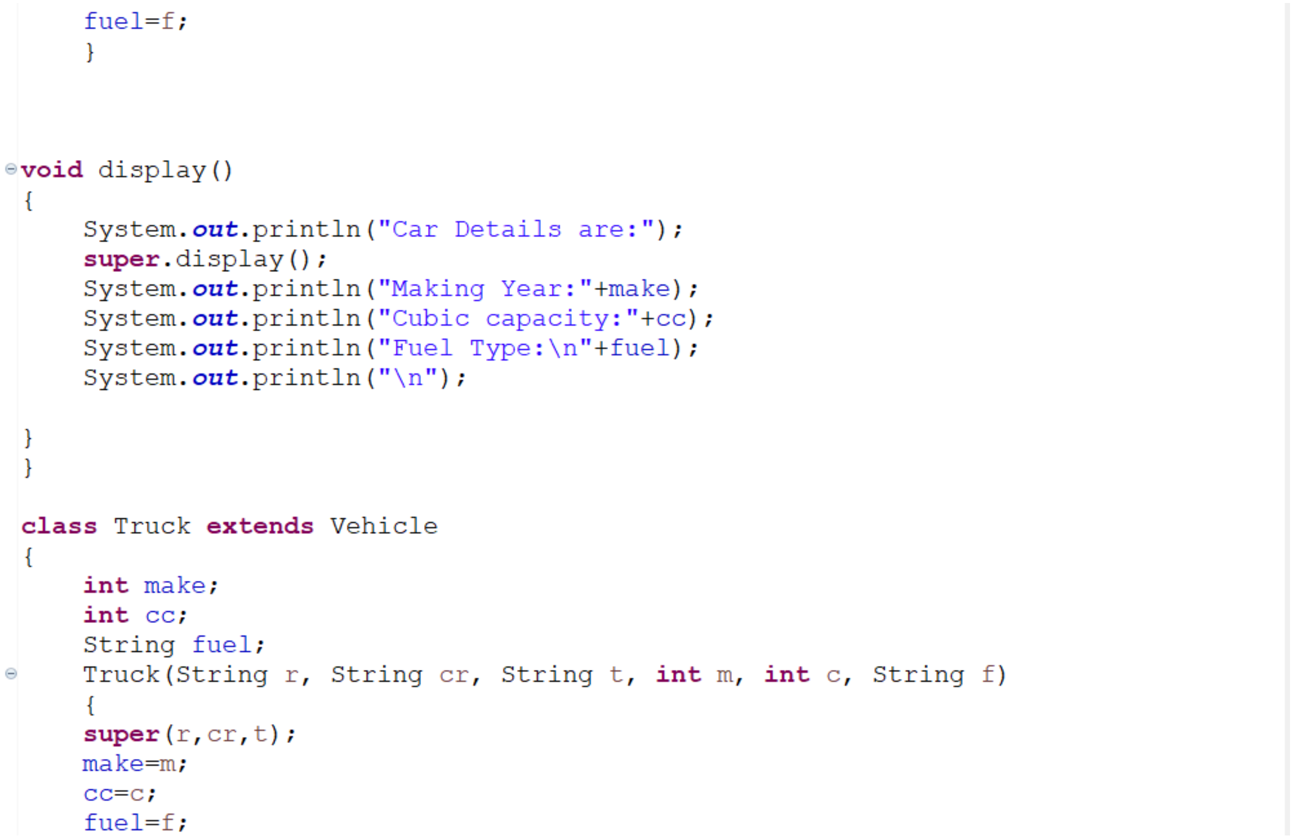
FAQs

1. Explain Inheritance?
2. Advantages of Inheritance?
3. What is parent class and child class?
4. Why does java not support Multiple inheritance?
5. Explain different types of inheritances?

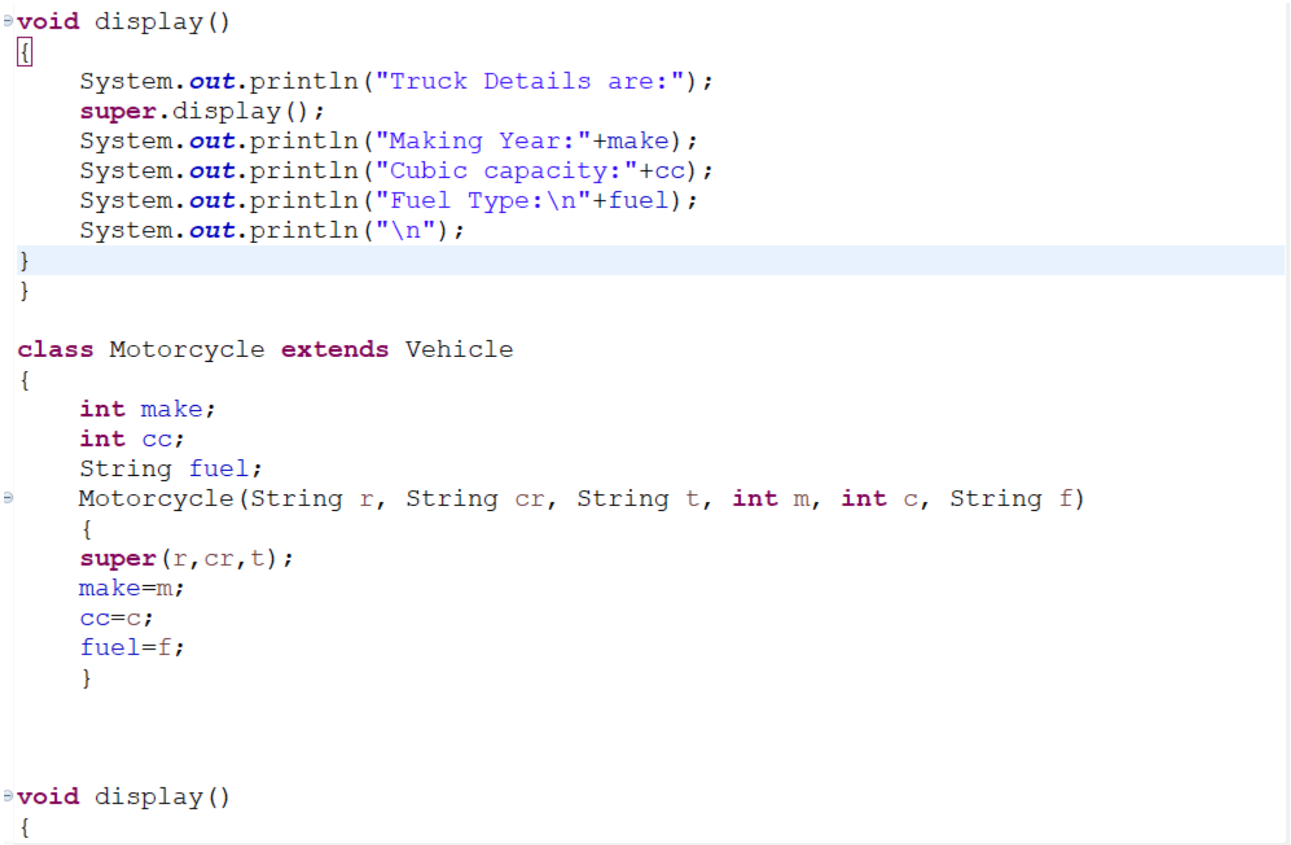
CODE: -



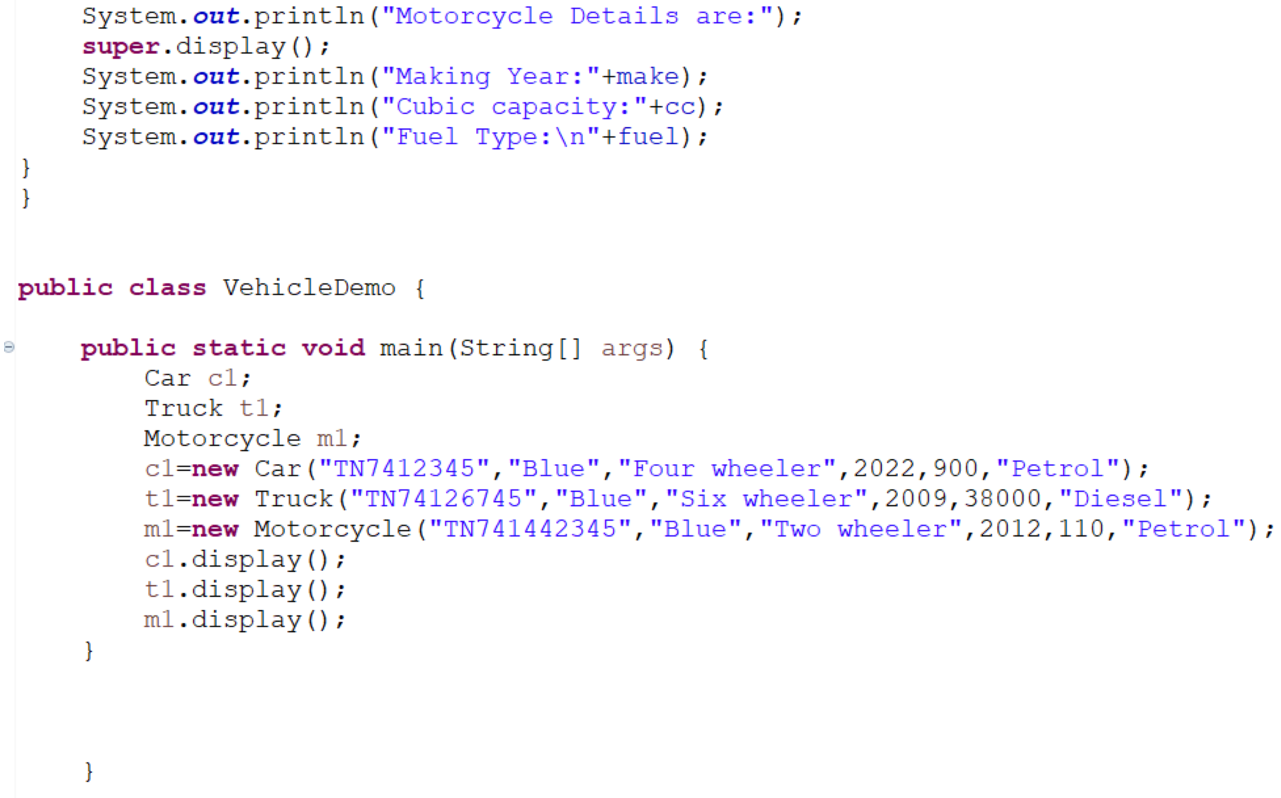
```
package unit2;
class Vehicle
{
    String regno;
    String color;
    String type_of_vehicle;
    Vehicle(String r,String cr,String t)
    {
        regno=r;
        color=cr;
        type_of_vehicle=t;
    }

void display()
{
    System.out.println("Registration no:"+regno);
    System.out.println("Color:"+color);
    System.out.println("Type:"+type_of_vehicle);
 }
}

class Car extends Vehicle
{
    int make;
    int cc;
    String fuel;
    Car(String r, String cr, String t, int m, int c, String f)
    {
    super(r,cr,t);
    make=m;
    cc=c;
```



```
    fuel=f;
    }



void display()
{
    System.out.println("Car Details are:");
    super.display();
    System.out.println("Making Year:"+make);
    System.out.println("Cubic capacity:"+cc);
    System.out.println("Fuel Type:\n"+fuel);
    System.out.println("\n");

}
}

class Truck extends Vehicle
{
    int make;
    int cc;
    String fuel;
    Truck(String r, String cr, String t, int m, int c, String f)
    {
    super(r,cr,t);
    make=m;
    cc=c;
    fuel=f;
```

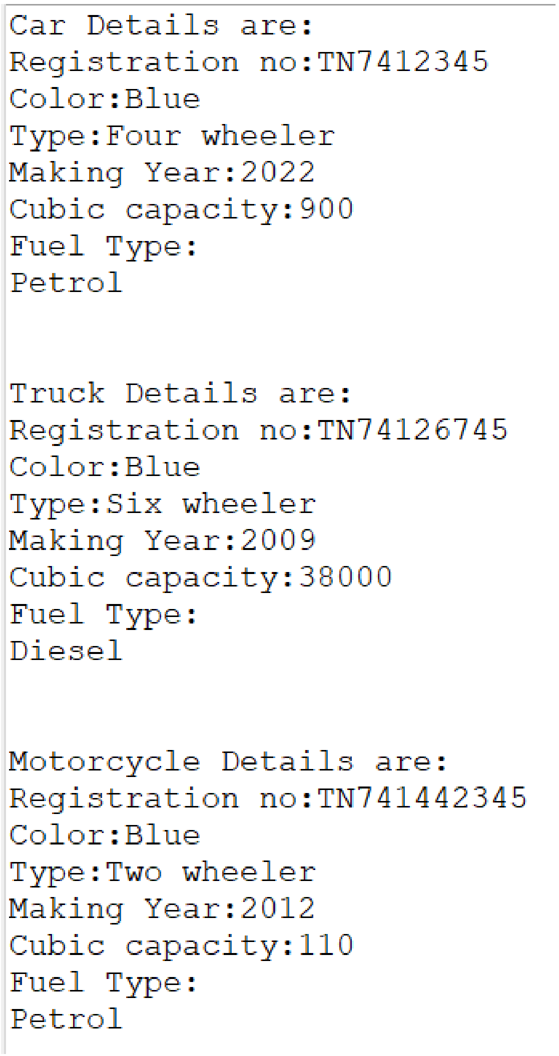


```java
void display()
{
    System.out.println("Truck Details are:");
    super.display();
    System.out.println("Making Year:"+make);
    System.out.println("Cubic capacity:"+cc);
    System.out.println("Fuel Type:\n"+fuel);
    System.out.println("\n");
}
}

class Motorcycle extends Vehicle
{
    int make;
    int cc;
    String fuel;
    Motorcycle(String r, String cr, String t, int m, int c, String f)
    {
    super(r,cr,t);
    make=m;
    cc=c;
    fuel=f;
    }



void display()
{
```



```java
    System.out.println("Motorcycle Details are:");
    super.display();
    System.out.println("Making Year:"+make);
    System.out.println("Cubic capacity:"+cc);
    System.out.println("Fuel Type:\n"+fuel);
}
}


public class VehicleDemo {

    public static void main(String[] args) {
        Car c1;
        Truck t1;
        Motorcycle m1;
        c1=new Car("TN7412345","Blue","Four wheeler",2022,900,"Petrol");
        t1=new Truck("TN74126745","Blue","Six wheeler",2009,38000,"Diesel");
        m1=new Motorcycle("TN741442345","Blue","Two wheeler",2012,110,"Petrol");
        c1.display();
        t1.display();
        m1.display();
    }



    }
```

OUTPUT:-



```
Car Details are:
Registration no:TN7412345
Color:Blue
Type:Four wheeler
Making Year:2022
Cubic capacity:900
Fuel Type:
Petrol


Truck Details are:
Registration no:TN74126745
Color:Blue
Type:Six wheeler
Making Year:2009
Cubic capacity:38000
Fuel Type:
Diesel


Motorcycle Details are:
Registration no:TN741442345
Color:Blue
Type:Two wheeler
Making Year:2012
Cubic capacity:110
Fuel Type:
Petrol
```