

1. Jelaskan apa yang dimaksud dengan class dalam pemrograman berorientasi objek. Berikan contoh sederhana untuk mengilustrasikan penggunaan class.

Class adalah blueprint atau cetakan untuk membuat objek. Class mendefinisikan atribut (variabel) dan metode (fungsi) yang akan dimiliki oleh objek yang dibuat berdasarkan class tersebut. Contoh sederhana class adalah "Mobil" memiliki atribut seperti "merek" dan "warna", serta metode seperti "jalan()" dan "berhenti()".

2. Jelaskan apa yang dimaksud dengan objek dalam pemrograman berorientasi objek. Berikan contoh sederhana untuk mengilustrasikan pembuatan dan penggunaan objek.

Objek adalah instansiasi atau perwujudan dari sebuah class. Dalam pemrograman berorientasi objek, objek adalah representasi konkret dari sebuah class yang memiliki atribut dan perilaku sesuai dengan yang telah didefinisikan dalam class tersebut.

```
// Person.java
class Person {
    String name;
    int age;

    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }
}

// Main.java
public class Main {
    public static void main(String[] args) {
        Person p1 = new Person("Cavin", 19);

        System.out.println("Nama: " + p1.name); // Nama: Cavin
        System.out.println("Umur: " + p1.age + " tahun"); // Umur: 19 tahun
    }
}
```

Source Code pembuatan dan penggunaan objek di Java

3. Jelaskan perbedaan antara class dan objek dalam pemrograman berorientasi objek.

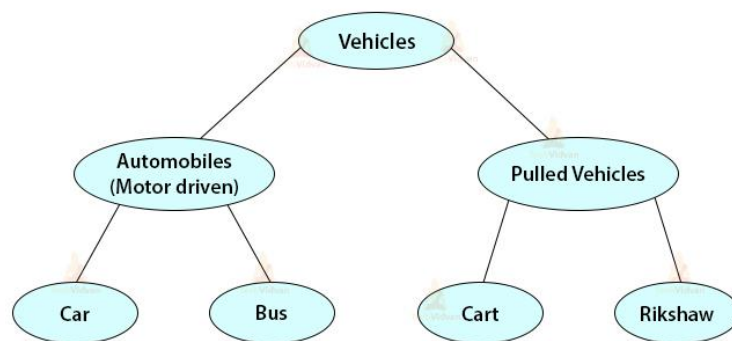
Class	Objek
Class adalah sebuah blueprint atau template yang mendefinisikan struktur, perilaku, dan karakteristik umum dari suatu objek.	Objek merupakan manifestasi atau hasil dari class yang telah diinstansiasi atau dibuat
Class digunakan untuk mendefinisikan variabel anggota (instance variables), metode, dan konstruktor yang akan dimiliki oleh objek-objek yang dibuat dari class tersebut.	Setiap objek yang dibuat dari class memiliki keunikannya sendiri dan dapat memiliki nilai yang berbeda untuk variabel anggota (instance variables) yang dimilikinya.
Class menyediakan kerangka dasar untuk menciptakan objek-objek yang memiliki atribut dan perilaku yang sama.	Objek adalah instansi konkret atau nyata dari suatu class

4. Jelaskan konsep enkapsulasi dalam pemrograman berorientasi objek. Mengapa enkapsulasi penting dalam pengembangan perangkat lunak?

Enkapsulasi adalah konsep dalam pemrograman berorientasi objek yang mengizinkan penyembunyian detail implementasi dari luar. Dengan menggunakan enkapsulasi, atribut suatu class dibungkus bersama metode-metode yang mengakses atau memanipulasi atribut tersebut. Ini membantu dalam mencapai konsep data hiding dan penggunaan class secara lebih aman. Enkapsulasi penting dalam pengembangan perangkat lunak karena meningkatkan keamanan dan modularitas, serta mengurangi ketergantungan antar bagian-bagian program.

5. Jelaskan konsep pewarisan (*inheritance*) dalam pemrograman berorientasi objek. Berikan contoh sederhana untuk mengilustrasikan pewarisan antara dua class.

Pewarisan adalah konsep di mana sebuah class dapat mewarisi atribut dan metode dari class lain yang disebut class induk atau superclass. Class yang mewarisi disebut subclass atau turunan. Dengan pewarisan, subclass dapat menggunakan atribut dan metode dari superclass dan juga dapat menambahkan atau memodifikasi perilaku dari superclass tersebut.



Gambar untuk mengilustrasikan pewarisan antara dua class