OOPS IN JAVA

1. What is the need for OOPs?

There are many reasons why OOPs is mostly preferred, but the most important among them are:

- OOPs helps users to understand the software easily, although they don't know the actual implementation.
- With OOPs, the readability, understandability, and maintainability of the code increase multifold.
- Even very big software can be easily written and managed easily using OOPs.

2. What are some advantages of using OOPs?

- OOPs is very helpful in solving very complex level of problems.
- Highly complex programs can be created, handled, and maintained easily using object-oriented programming.
- OOPs, promote code reuse, thereby reducing redundancy.
- OOPs also helps to hide the unnecessary details with the help of Data Abstraction.
- Polymorphism offers a lot of flexibility in OOPs.

3. What are some major Object Oriented Programming languages?

Some of the major Object-Oriented Programming languages include : **Java** , **C++** , **Javascript** , **Python** ,**PHP** , And many more.

4. What is meant by Structured Programming?

Structured Programming refers to the method of programming which consists of a completely structured control flow. Here structure refers to a block, which contains a set of rules, and has a definitive control flow, such as (if/then/else), (while and for), block structures, and subroutines.

Nearly all programming paradigms include Structured programming, including the OOPs model.

5. What are the main features of OOPs?

OOPs or Object Oriented Programming mainly comprises of the below four features, and make sure you don't miss any of these:

- Inheritance
- Encapsulation
- Polymorphism
- Data Abstraction

6. What is a class?

A class can be understood as a template or a blueprint, which contains some values, known as member data or member, and some set of rules, known as behaviors or functions. So when an object is created, it automatically takes the data and functions that are defined in the class.

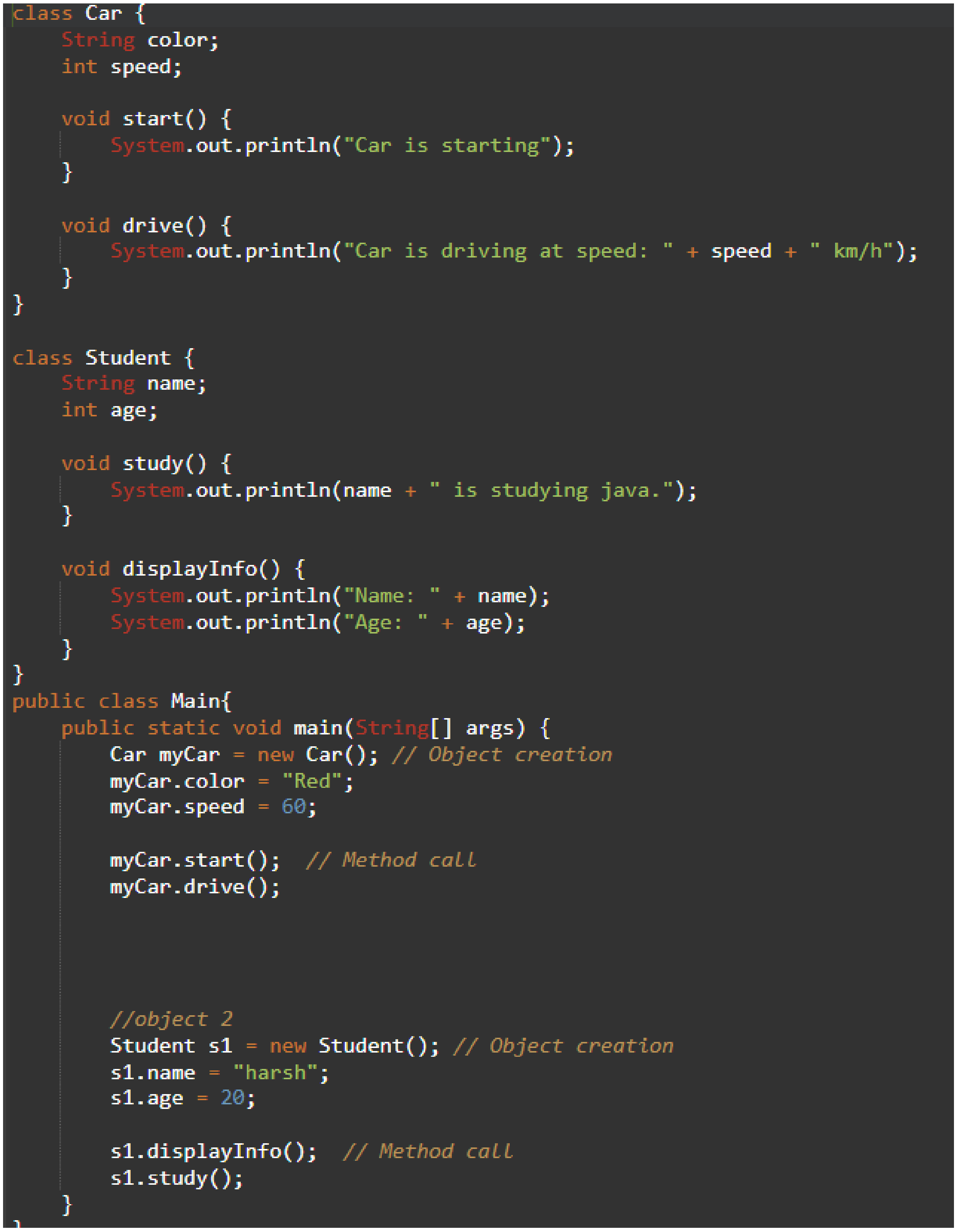
For example, first, a car's template is created. Then multiple units of car are created based on that template.

7. What is an object?

An object refers to the instance of the class, which contains the instance of the members and behaviors defined in the class template. In the real world, an object is an actual entity to which a user interacts, whereas class is just the blueprint for that object. So the objects consume space and have some characteristic behavior. For example, a specific car.

8. How much memory does a class occupy?

Classes do not consume any memory. They are just a blueprint based on which objects are created. Now when objects are created, they actually initialize the class members and methods and therefore consume memory.



```java
class Car {
    String color;
    int speed;

    void start() {
        System.out.println("Car is starting");
    }

    void drive() {
        System.out.println("Car is driving at speed: " + speed + " km/h");
    }
}

class Student {
    String name;
    int age;

    void study() {
        System.out.println(name + " is studying java.");
    }

    void displayInfo() {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
    }
}
public class Main{
    public static void main(String[] args) {
        Car myCar = new Car(); // Object creation
        myCar.color = "Red";
        myCar.speed = 60;

        myCar.start();  // Method call
        myCar.drive();



        //object 2
        Student s1 = new Student(); // Object creation
        s1.name = "harsh";
        s1.age = 20;

        s1.displayInfo();  // Method call
        s1.study();
    }
```

9. Java this Keyword

The this keyword in Java refers to the current object in a method or constructor.

The this keyword is often used to avoid confusion when class attributes have the same name as method or constructor parameters.

8. What is a constructor?

Constructors are special methods whose name is the same as the class name.

A constructor in Java is a **special method** that is used to initialize objects.

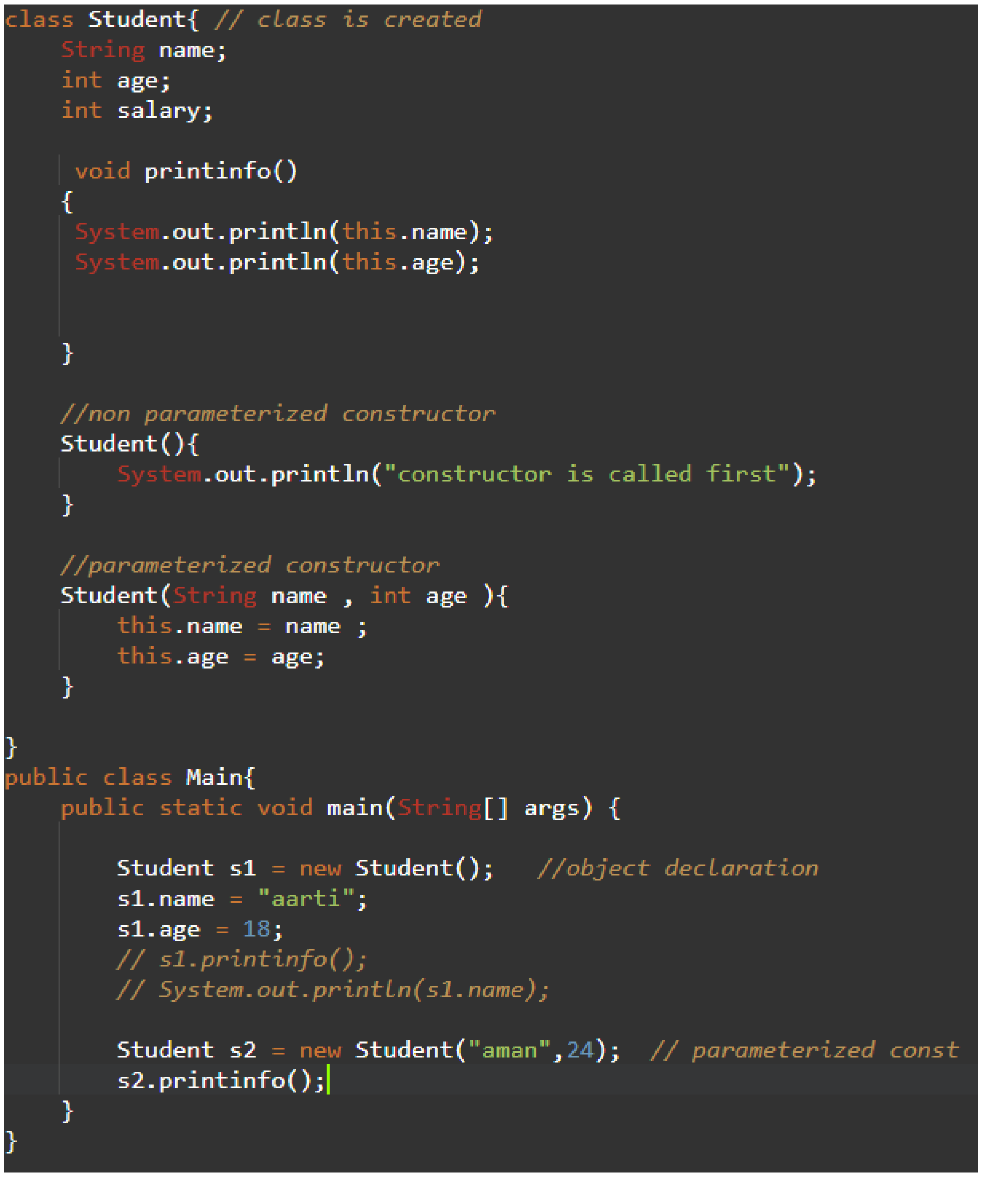
The constructor is called when an object of a class is created.

9. What are the various types of constructors in C++?

The most common classification of constructors includes:

Default constructor: The default constructor is the constructor which doesn't take any argument. It has no parameters

Parameterized constructor: The constructors that take some arguments are known as parameterized constructors.



```java
class Student{ // class is created
    String name;
    int age;
    int salary;

     void printinfo()
    {
     System.out.println(this.name);
     System.out.println(this.age);


    }

    //non parameterized constructor
    Student(){
        System.out.println("constructor is called first");
    }

    //parameterized constructor
    Student(String name , int age ){
        this.name = name ;
        this.age = age;
    }

}
public class Main{
    public static void main(String[] args) {

        Student s1 = new Student();   //object declaration
        s1.name = "aarti";
        s1.age = 18;
        // s1.printinfo();
        // System.out.println(s1.name);

        Student s2 = new Student("aman",24);  // parameterized const
        s2.printinfo();
    }
}
```

10. Encapsulation

Encapsulation in Java is one of the core concepts of Object-Oriented Programming (OOP). It refers to wrapping the data (variables) and code (methods) together as a single unit, and restricting direct access to some of the object's components.

This is usually done using:

- Private variables
- Public getter and setter methods

11. What are access specifiers and what is their significance?

Access specifiers, as the name suggests, are a special type of keywords, which are used to control or specify the accessibility of entities like classes, methods, etc. Some of the access specifiers or access modifiers include “private”, “public”, etc. These access specifiers also play a very vital role in achieving Encapsulation - one of the major features of OOPs

12. Explain Inheritance with an example?

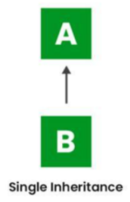
Inheritance is one of the major features of object-oriented programming, by which an entity inherits some characteristics and behaviors of some other entity and makes them their own. Inheritance helps to improve and facilitate code reuse.

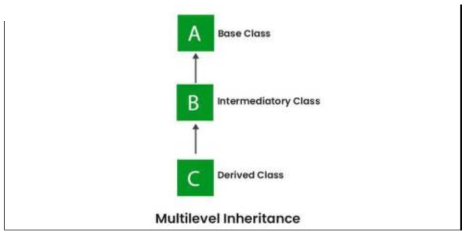
Example : Let's take three different vehicles - a car, truck, or bus. These three are entirely different from one another with their own specific characteristics and behavior. But. in all three, you will find some common elements, like steering wheel, accelerator, clutch, brakes, etc. Though these elements are used in different vehicles, still they have their own features which are common among all vehicles. This is achieved with inheritance. The car, the truck, and the bus have all inherited the features like steering wheel, accelerator, clutch, brakes, etc, and used them as their own. Due to this, they did not have to create these components from scratch, thereby facilitating code reuse

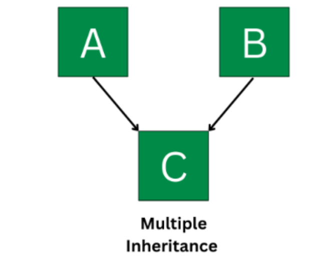
13. What are the various types of inheritance?

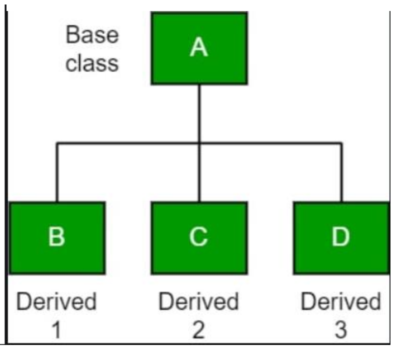
The various types of inheritance include:

- Single inheritance
- Multiple inheritances
- Multi-level inheritance
- Hierarchical inheritance
- Hybrid inheritance









14. What is a subclass? The subclass is a part of Inheritance. The subclass is an entity, which inherits from another class. It is also known as the child class.

Define a superclass? Superclass is also a part of Inheritance. The superclass is an entity, which allows subclasses or child classes to inherit from itself.

15. What is Polymorphism?

Polymorphism is composed of two words - "poly" which means "many", and "morph" which means "shapes". Therefore Polymorphism refers to something that has many shapes.

In OOPs, Polymorphism refers to the process by which some code, data, method, or object behaves differently under different circumstances or contexts. Compile-time polymorphism and Run time polymorphism are the two types of polymorphisms in OOPs languages.

16. What is meant by static polymorphism?

Static Polymorphism is commonly known as the Compile time polymorphism. Static polymorphism is the feature by which an object is linked with the respective function or operator based on the values during the compile time. Static or Compile time Polymorphism can be achieved through Method overloading or operator overloading.

17. What is meant by dynamic polymorphism?

Dynamic Polymorphism or Runtime polymorphism refers to the type of Polymorphism in OOPs,

by which the actual implementation of the function is decided during the runtime or execution. The dynamic or runtime polymorphism can be achieved with the help of method overriding.

18. What is the difference between overloading and overriding?

Overloading is a compile-time polymorphism feature in which an entity has multiple implementations with the same name. For example, Method overloading and Operator overloading.

Whereas Overriding is a runtime polymorphism feature in which an entity has the same name, but its implementation changes during execution. For example, Method overriding.

19. What is an exception?

An exception can be considered as a special event, which is raised during the execution of a program at runtime, that brings the execution to a halt. The reason for the exception is mainly due to a position in the program, where the user wants to do something for which the program is not specified, like undesirable input.

20. What is meant by exception handling?

No one wants its soware to fail or crash. Exceptions are the major reason for soware failure. The exceptions can be handled in the program beforehand and prevent the execution from stopping. This is known as exception handling. So exception handling is the mechanism for identifying the undesirable states that the program can reach and specifying the desirable outcomes of such states. Try-catch is the most common method used for handling exceptions in the program.

21. What is meant by Garbage Collection in OOPs world?

Object-oriented programming revolves around entities like objects. Each object consumes memory and there can be multiple objects of a class. So if these objects and their memories are not handled properly, then it might lead to certain memoryrelated errors and the system might fail. Garbage collection refers to this mechanism of handling the memory in the program.

Through garbage collection, the unwanted memory is freed up by removing the objects that are no longer needed.