Week3-Session 1 Interview Questions

1. **What is Inheritance in Java?**

Inheritance is an Object-oriented feature which allows a class to inherit behavior and data from other classes. For example, a class Car can extend the basic feature of Vehicle class by using Inheritance. One of the most intuitive examples of Inheritance in the real world is Father-Son relationship, where Son inherits Father's properties and behavior.

2. **Which types of inheritance are supported by Java?**

Java supports single, multi-level, and hierarchical inheritance. It does not directly support multiple and hybrid inheritance. Single inheritance is where a subclass extends a single superclass, while multi-level inheritance is where a subclass extends a superclass, which in turn extends another superclass. Hierarchical inheritance is where multiple subclasses extend a single superclass.

Multiple inheritances is where a subclass extends multiple super classes, which is not supported in Java. Hybrid inheritance is a combination of two or more types of inheritance.

3**. How can we achieve multiple inheritances in Java?**

Java does not directly support multiple inheritance of classes, meaning that a class cannot inherit from more than one class at a time. However, Java does support multiple inheritance of interfaces, where a class can implement multiple interfaces. This allows a class to inherit the abstract behavior defined in each interface, providing a way to achieve the benefits of multiple inheritance without the complications of inheriting from multiple classes.

4. **Why is multiple inheritance not supported in java?**

To reduce the complexity and simplify the language, multiple inheritance is not supported in java. Consider a scenario where A, B, and C are three classes. The C class inherits A and B classes. If A and B classes have the same method named display(), and you call it from a child class object, there will be ambiguity when calling the method of A or B class.

5. **Why is Inheritance used in Java?**

There are various advantages of using inheritance in Java as given below.

- Inheritance provides code reusability. The derived class does not need to redefine the method of base class unless it needs to provide the specific implementation of the method.
- Runtime polymorphism cannot be achieved without using inheritance. ● We can simulate the inheritance of classes with the real-time objects which makes OOPs more realistic.

- Inheritance provides data hiding. The base class can hide some data from the derived class by making it private.
- Method overriding cannot be achieved without inheritance. By method overriding, we can give a specific implementation of some basic method contained by the base class.

6. **What is method overloading?**

Method overloading is the polymorphism technique which allows us to create multiple methods with the same name but different signatures. We can achieve method overloading in two ways.

o By Changing the number of arguments

o By Changing the data type of arguments

Method overloading increases the readability of the program. Method overloading is performed to figure out the program quickly.

7. Can we overload the methods by making them static?

No, we cannot overload the methods by just applying the static keyword to them (number of parameters and types are the same).

8. Can we overload the main () method?

Yes, we can have any number of main methods in a Java program by using method overloading.

9. What is method overriding?

If a subclass provides a specific implementation of a method that is already provided by its parent class, it is known as Method Overriding. It is used for runtime polymorphism and to implement the interface methods.

Rules for Method overriding

o The method must have the same name as in the parent class.

o The method must have the same signature as in the parent class.

o Two classes must have an IS-A relationship between them.

10. Can we override the static method?

No, you can't override the static method because they are the part of the class, not the object.

11. Why can we not override the static method?

It is because the static method is the part of the class, and it is bound with the class whereas the instance method is bound with the object, and static gets memory in class area, and instance gets memory in a heap.

12. **Can we override the overloaded method?**

Yes.

13. **Difference between method Overloading and Overriding.**

Method Overloading	**Method Overriding**
1) Method overloading increases the readability of the program.	Method overriding provides the specific implementation of the method that is already provided by its superclass.
2) Method overloading occurs within the class.	Method overriding occurs in two classes that have IS-A relationship between them.
3) In this case, the parameters must be different.	In this case, the parameters must be the same.

14. **Can we override the private methods?**

No, we cannot override the private methods because the scope of private methods is limited to the class and we cannot access them outside of the class.

15. **Which class is the superclass for all the classes?**

The Object class is the superclass of all other classes in Java.

16. **What is the difference between Inheritance and Encapsulation?**

Inheritance is an object oriented concept which creates a parent-child relationship. It is one of the ways to reuse the code written for parent class but it also forms the basis of Polymorphism. On the other hand, Encapsulation is an object oriented concept which is used to hide the internal details of a class e.g. Hash Map encapsulate how to store elements and how to calculate hash values.

17. **What is the difference between Inheritance and Abstraction?**

Abstraction is an object oriented concept which is used to simplify things by abstracting details. It helps in the designing system. On the other hand, Inheritance allows code reuse. You can reuse the functionality you have already coded by using Inheritance.