**DAY 20**

**INTERFACE**

Interfaces

All the fields in interfaces are public, static and final by default.

All methods are public & abstract by default.

A class that implements an interface must implement all the methods declared in the interface.

Interfaces support the functionality of multiple inheritance.

**Q) Multiple inheritance is not supported through class in java, but it is possible by an interface, why?**

As we have explained in the inheritance chapter, multiple inheritance is not supported in the case of class because of ambiguity. However, it is supported in case of an interface because there is no ambiguity. It is because its implementation is provided by the implementation class. For example:

interface Printable{

void print();

}

interface Showable{

void print();

}

class TestInterface3 implements Printable, Showable{

public void print(){System.out.println("Hello");}

public static void main(String args[]){

TestInterface3 obj = new TestInterface3();

obj.print();

}

}

Output:

Hello

As you can see in the above example, Printable and Showable interface have same methods but its implementation is provided by class TestTnterface1, so there is no ambiguity.

**QUESTION**

(1) What is interface in java?

Java interface is a blueprint of a class and it is used to achieve fully abstraction and it is a collection of abstract methods.

(2) Can we achieve multiple inheritance by using interface?

Yes, we can achieve multiple inheritance through the interface in java but it is not possible through class because java doesn't support multiple inheritance through class.

(3) How to declare an interface, write a syntax?

The syntax of declaring the interface in java by using interface keyword :

interface FirstInterface

{

fields;

methods;

}

(4) Can we create an object of an interface?

No, we cannot create an object of interface.

(5) Can we declare the interface as final?

No, we can't declare the interface as final because the implementation of the interface is provided by another class. If we make the interface as final, it will throw a compile-time error.

(6) Which keyword java compiler add before interface fields and methods?

In an interface, Java compiler adds public, static and final keywords before fields or data members and add public abstract keywords before methods. In other words, all the fields are public, static and final and all the methods are public and abstract by default in an interface.

(7) Does interface extend Object class by default?

No, Interface does not extend Object class in java by default but all the classes extend Object class by default.

(8) Can an interface extend another interface?

Yes, an interface can extend another interface.

(9) Can an interface extend a class?

No, A class can implement an interface but interface cannot extend a class.

(10) Can we put a static method in interfaces?

No, we cannot put static methods in interfaces because all the methods are by default public and abstract in the interface and we cannot use abstract and static keywords together.

(11) Can we declare an interface with the abstract keyword?

Yes, we can declare abstract keyword with interfaces but there is no need to write abstract keyword with interfaces because all the interfaces are abstract by default.

(12) What is default keyword in an interface?

By the help of default keyword, we can keep non-abstract method in java interface i.e with method body{}. This is the new feature of JAVA 8.

Syntax of default keyword :

interface Test

{

default void show()

{

-----

-----

}

}

(13) What is marker or tagged interface?

Marker interface is an interface that has no data member and method like Serializable, Cloneable, etc.

(14) Can we declare a constructor in the interface?

No.

(15) After compilation of interface program, .class file will be generated for every interface in java... true or false.?

This is true.

(16) Can we change the value of a field in interface after initialization?

No, Because all the fields of the interface are by default final.

(17) Difference between abstract class and interface?

There are many differences between abstract class and interface.

abstract class

Through abstract class, you cannot achieve multiple inheritance.

You can keep non-abstract method(with method body) in the abstract class.

In the abstract class, fields are not public, static, final and methods are not public abstract by default.

interface

Through the interface, you can achieve multiple inheritance.

In an interface, you can't keep non-abstract method but since java 8 it is possible.

In an interface, fields are public, static, and final and methods are public and abstract by default.