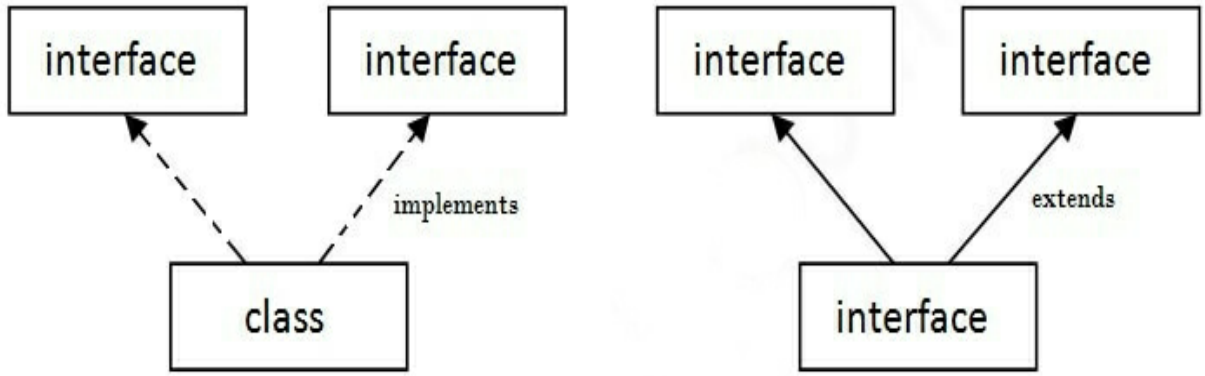
BCA – 401: Java Programming

Rahul Kumar Singh

In today's Class we have discussed on Interface in Java.

Multiple inheritance in Java by interface:-

If a class implements multiple interfaces, or an interface extends multiple interfaces, it is known as multiple inheritance.



Multiple Inheritance in Java

Example:-

```
interface Printable
{
void print();
}
interface Showable
{
```

```
void show();
}
class A implements Printable,Showable
{
public void print()
{
System.out.println("Hello");
}
public void show()
{
System.out.println("Welcome");
}
  public static void main(String args[])
{
A obj = new A();
obj.print();
obj.show();
 }
}
```

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.

Java 8 Default Method in Interface:-

Since Java 8, we can have method body in interface. But we need to make it default method.

Example:-

```
interface Drawable
{
void draw();
default void msg()
```

```
{
System.out.println("default method");
}
}
class Rectangle implements Drawable
{
public void draw()
{
System.out.println("drawing rectangle");
}
}
class TestInterfaceDefault
{
public static void main(String args[])
{
Drawable d=new Rectangle();
d.draw();
d.msg();
}
}
```

Output:-

drawing rectangle

default method

Java 8 Static Method in Interface:-

Since Java 8, we can have static method in interface.

Example:-

```
interface Drawable
{
void draw();
static int cube(int x)
{
return x*x*x;
}
}
class Rectangle implements Drawable
{
public void draw()
{
System.out.println("drawing rectangle");
```

```
}
}
class TestInterfaceStatic
{
public static void main(String args[])
{
Drawable d=new Rectangle();
d.draw();
System.out.println(Drawable.cube(3));
}
}
```

Output:

```
drawing rectangle
27
```

Q) What is marker or tagged interface?

An interface which has no member is known as a marker or tagged interface, for example, Serializable, Cloneable, Remote, etc. They are used to provide some essential information to the JVM so that JVM may perform some useful operation.

Example:-//How Serializable interface is written?

```
public interface Serializable
{
}
```

Nested Interface in Java:-

An interface can have another interface which is known as a nested interface.

Example:-

```
interface printable
{
 void print();
 interface MessagePrintable
{
   void msg();
 }
}
```

Difference between abstract class and interface:-

Abstract class and interface both are used to achieve abstraction where we can declare the abstract methods. Abstract class and interface both can't be instantiated.

But there are many differences between abstract class and

interface that are given below.

S.no	Abstract class	Interface
01.	Abstract class can have abstract and non-abstract methods.	Interface can have only abstract methods. Since Java 8, it can have default and static methods also.
02.	Abstract class doesn't support multiple inheritance.	Interface supports multiple inheritance.
03.	Abstract class can have final, non-final, static and non-static variables.	Interface has only static and final variables.
04.	Abstract class can provide implementation of interface.	Interface can't provide the implementation of abstract class.
05.	The abstract keyword is used to declare abstract class.	The interface keyword is used to declare interface.
06.	An abstract class can extend another Java class and implement multiple Java interfaces.	An interface can extend another Java interface only.
07.	An abstract class can be extended using keyword	An interface can be implemented using

	"extends".	keyword "implements".
08.	A Java abstract class can have class members like private, protected, etc.	Members of a Java interface are public by default.

09.	**Example:**
	public abstract class Shape
	{
	public abstract void draw();
	}
		Example:
		public interface Drawable
		{
		void draw();
		}

Simply, abstract class achieves partial abstraction (0 to 100%) whereas interface achieves fully abstraction (100%).

Example of abstract class and interface in Java

In following example we are using interface and abstract class both.

```
//Creating interface that has 4 methods

interface A

{

void a();   //by default, public and abstract
```

```
void b();

void c();

void d();

}

//Creating abstract class that provides the implementation of one method of A interface

abstract class B implements A

{

public void c()

{

System.out.println("I am C");

}

}

//Creating subclass of abstract class, now we need to provide the implementation of rest of the methods

class M extends B

{

public void a()

{

System.out.println("I am a");

}
```

```
public void b()
{
System.out.println("I am b");
}
public void d()
{
System.out.println("I am d");
}
}
//Creating a test class that calls the methods of A interface
class Test
{
public static void main(String args[])
{
A a=new M();
a.a();
a.b();
a.c();
a.d();
}
```

```
}
```

Output:

```
I am a
I am b
I am c
I am d
```