BCA – 401: Java Programming

Rahul Kumar Singh

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

Defining/Declaring an interface:-

An interface is declared by using the interface keyword. It provides total abstraction; means all the methods in an interface are declared with the empty body, and all the fields are public, static and final by default. A class that implements an interface must implement all the methods declared in the interface.

Syntax:

```
interface <interface_name>

{

   // declare constant fields

   // declare methods that abstract by default.

}
```

Interfaces have the following properties:–

- An interface is implicitly abstract. You do not need to use the abstract keyword while declaring an interface.
- Each method in an interface is also implicitly abstract, so the abstract keyword is not needed.
- Methods in an interface are implicitly public.

Example:-

```
/* File name : Animal.java */
interface Animal
{
   public void eat();
   public void travel();
}
```

Implementing Interfaces:-

When a class implements an interface, you can think of the class as signing a contract, agreeing to perform the specific behaviors of the interface. If a class does not perform all the behaviors of the interface, the class must declare itself as abstract.

A class uses the implements keyword to implement an interface. The implements keyword appears in the class declaration following the extends portion of the declaration.

Example:-

```
/* File name : MammalInt.java */
public class MammalInt implements Animal
{
   public void eat()
   {
       System.out.println("Mammal eats");
   }
   public void travel()
   {
       System.out.println("Mammal travels");
   }
   public int noOfLegs()
   {
       return 0;
   }
   public static void main(String args[])
   {
       MammalInt m = new MammalInt();
       m.eat();
       m.travel();
   }
}
```

Output:-

```
Mammal eats
Mammal travels
```

When overriding methods defined in interfaces, there are several rules to be followed –

- Checked exceptions should not be declared on implementation methods other than the ones declared by the interface method or subclasses of those declared by the interface method.
- The signature of the interface method and the same return type or subtype should be maintained when overriding the methods.
- An implementation class itself can be abstract and if so, interface methods need not be implemented.

When implementation interfaces, there are several rules –

- A class can implement more than one interface at a time.
- A class can extend only one class, but implement many interfaces.
- An interface can extend another interface, in a similar way as a class can extend another class.

Java Interface Example:-

In this example, the Printable interface has only one method, and its implementation is provided in the A class.

```
interface printable
{
void print();
}
class A implements printable
{
public void print()
{
System.out.println("Hello");
}
public static void main(String args[])
{
A obj = new A();
obj.print();
 }
}
```

Output:

```
Hello
```

Extending Interfaces:-

An interface can extend another interface in the same way that a class can extend another class. The **extends** keyword is used to extend an interface, and the child interface inherits the methods of the parent interface.

A class implements an interface, but one interface extends another interface.

The following Sports interface is extended by Hockey and Football interfaces.

Example:-

```
// Filename: Sports.java
public interface Sports
{
   public void setHomeTeam(String name);
   public void setVisitingTeam(String name);
}
// Filename: Football.java
public interface Football extends Sports
{
   public void homeTeamScored(int points);
   public void visitingTeamScored(int points);
   public void endOfQuarter(int quarter);
}
// Filename: Hockey.java
public interface Hockey extends Sports
{
   public void homeGoalScored();
   public void visitingGoalScored();
   public void endOfPeriod(int period);
   public void overtimePeriod(int ot);
}
```

The Hockey interface has four methods, but it inherits two from Sports; thus, a class that implements Hockey needs to implement all six methods. Similarly, a class that implements Football needs to define the three methods from Football and the two methods from Sports.

Example-2

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

```
interface Showable extends Printable
{
void show();
}
class TestInterface4 implements Showable
{
public void print()
{
System.out.println("Hello");
}
public void show()
{
System.out.println("Welcome");
}
public static void main(String args[])
{
TestInterface4 obj = new TestInterface4();
obj.print();
obj.show();
 }
}
```

Output:

```
Hello

Welcome
```