# **Interfaces**

**Define interface in java?**

Interface is collection of abstract methods and constants. An interface is also defined as pure or 100 percent abstract class. Interfaces are implicitly abstract whether we define abstract access modifier or not. A class implementing interface overrides all the abstract methods defined in interface. Implements keyword is used to implement interface.

**What is the purpose of interface?**

Interface is a contract. Interface acts like a communication between two objects. When we are defining an interface, we are defining a contract what our class should do but not how it does. An interface doesn’t define what a method does. The power of interface lies when different classes that are unrelated can implement interface. Interfaces are designed to support dynamic method resolution at run time.

**Explain features of interfaces in java?**

1) All the methods defined in interfaces are implicitly abstract even though abstract modifier is not declared.

2) All the methods in interface are public whether they are declared as public or not.

3) variables declared inside interface are by default public, static and final.

4) Interfaces cannot be instantiated.

5) we cannot declare static methods inside interface.

6) ‘implements’ keyword is used to implement interface.

7) Unlike class, interface can extend any number of interfaces.

8) We can define a class inside interface and the class acts like inner class to interface.

9) An interface can extend a class and implement an interface

10) Multiple inheritance in java is achieved through interfaces.

**Class C implements Interface I containing method m1 and m2 declarations. Class C has provided implementation for method m2. Can i create an object of Class C?**

No not possible. Class C should provide implementation for all the methods in the Interface I. Since Class C didn't provide implementation for m1 method, it has to be declared as abstract. Abstract classes can't be instantiated.

**Can a method inside a Interface be declared as final?**

No not possible. Doing so will result in compilation error. public and abstract are the only applicable modifiers for method declaration in an interface.

**Can an Interface implement another Interface?**

Interfaces don't provide implementation hence an interface cannot implement another interface. Interface extends another interface.

**Can an Interface extend another Interface?**

Yes an Interface can inherit another Interface, for that matter an Interface can extend more than one Interface.

**Can a Class extend more than one Class?**

Not possible. A Class can extend only one class but can implement any number of Interfaces.

**Why is an Interface be able to extend more than one Interface but a Class can't extend more than one Class?**

Basically Java doesn't allow multiple inheritance, so a Class is restricted to extend only one Class. But an Interface is a pure abstraction model and doesn't have inheritance hierarchy like classes(do remember that the base class of all classes is Object). So an Interface is allowed to extend more than one Interface.

**Can an Interface be final?**

Not possible. Doing so so will result in compilation error.

**Can we define private and protected modifiers for variables in interfaces?**

No.

**What modifiers are allowed for methods in an Interface?**

Only public and abstract modifiers are allowed for methods in interfaces.

**When can an object reference be cast to an interface reference?**

An object reference be cast to an interface reference when the object implements the referenced interface.

**Can we define static methods inside interface?**

We can’t declare static methods inside interface. Only instance methods are permitted in interfaces. Only public and abstract modifiers are permitted for interface methods. If we try to declare static methods inside interface we get compilation error saying “Illegal modifier for the interface method Classname.methodName(); only public & abstract are permitted”.

**Difference between abstract class and interface?**

|  |  |
| --- | --- |
| **Interface** | **Abstract Class** |
| Interface contains only abstract methods | Abstract class contains abstract methods, concrete methods or both |
| Access Specifiers for methods in interface must be public | Except private we can have any access specifier for methods in abstract class. |
| Variables defined must be public , static , final | Except private variables can have any access specifiers |
| Multiple Inheritance in java is implemented using interface | We cannot achieve multiple inheritance using abstract class. |
| To implement an interface we use implements keyword | To extend an abstract class we use extends keyword. |