Can you use this() and super() both in a constructor?

No. Because super() or this() must be the first statement.

### Can we override static method?

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

### Why we cannot override static method?

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

### Can we override the overloaded method?

Yes.

### Can you declare the main method as final?

Yes, such as, public static final void main(String[] args){}.

### Can there be any abstract method without abstract class?

No, if there is any abstract method in a class, that class must be abstract.

### Can you declare an interface method static?

No, because methods of an interface is abstract by default, and static and abstract keywords can't be used together.

### Can an Interface be final?

No, because its implementation is provided by another class.

What is difference between abstract class and interface?

|  |  |
| --- | --- |
| **Abstract class** | **Interface** |
| 1)An abstract class can have method body (non-abstract methods). | Interface have only abstract methods. |
| 2)An abstract class can have instance variables. | An interface cannot have instance variables. |
| 3)An abstract class can have constructor. | Interface cannot have constructor. |
| 4)An abstract class can have static methods. | Interface cannot have static methods. |
| 5)You can extends one abstract class. | You can implement multiple interfaces. |

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

No, they are implicitly public.

68) When can an object reference be cast to an interface reference?

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

We cannot override static methods. Static methods are belogs to class, not belongs to object. Inheritance will not be applicable for class members