



which class can not extend another class in java



Sign in

[All](#) [Videos](#) [News](#) [Images](#) [Maps](#) [More](#) [Settings](#) [Tools](#)

About 8,67,00,000 results (0.76 seconds)

Did you mean: [which class **cannot** extend another class in java](#)

According to Java **Object superclass**, java.lang.Object does not extend Object . All classes implicitly extend Object if no other **superclass** is given. **Class Object** is the root of the class hierarchy. Every class has Object as a **superclass**. Oct 2, 2012

[java - Classes that don't inherit Object class - Stack Overflow](#)
<https://stackoverflow.com/questions/12684305/classes-that-dont-inherit-object-class>

[?](#) [About this result](#) [Feedback](#)

People also ask

Can you extend two classes in Java?

Java doesn't have **multiple** inheritance. **Java** does not support **multiple** inheritance, but **you can** try to implement **two** or more interface. ... **Java** does not allow **extending** from several **classes**. What **you** want is probably public **class** Main **extends** ListActivity implements ControlMenu . Apr 30, 2011

[java - Extending from two classes - Stack Overflow](#)
<https://stackoverflow.com/questions/5836662/extending-from-two-classes>

Search for: [Can you extend two classes in Java?](#)

Which classes Cannot be extended in Java?

A class declared as final cannot be extended while a method declared as final cannot be overridden in its **subclasses**. A method or a class is declared to be final using the **final keyword**. Though a final class cannot be extended, it can extend other classes.

[Final Classes and Methods - Java Tutorial - Java With Us](#)
www.javawithus.com/tutorial/final-classes-and-methods

Search for: [Which classes Cannot be extended in Java?](#)

Can a final class extend another class?

You **can't extend** a **class** that is declared **final** . However you **can** create an intermediate **class**, called Wrapper . This Wrapper **will** basically contain an instance of the original **class** and provide alternative methods to modify the state of the object according what you want to do. Nov 9, 2012

[static - How to extend a final class in Java - Stack Overflow](#)
<https://stackoverflow.com/questions/13294629/how-to-extend-a-final-class-in-java>

Search for: [Can a final class extend another class?](#)

Can enum extend any class Java?

Answer: **Enum can not extend any class in java**, the reason is by default, **Enum** extends abstract base **class java.lang.Enum**. Since **java does** not support multiple inheritance for **classes**, **Enum can not extend** another **class**.

[Can Enum extend any class in Java? - Java Interview Questions ...](#)
www.java2novice.com/java_interview_questions/can-enum-extend/

Search for: [Can enum extend any class Java?](#)

How many classes we can extend in Java?

Java does not support multiple inheritance, that's why you can't extend a class from **two** different classes at the same time. Rather, use a single class to extend from,

and use interfaces to include additional functionality. Jul 6, 2011

[java - Can one class extend two classes? - Stack Overflow](#)

<https://stackoverflow.com/questions/6587621/can-one-class-extend-two-classes>

Search for: [How many classes we can extend in Java?](#)

Can a class extend multiple abstract classes?

Basically, the rule says that you **can** inherit from (**extend**) as many **classes** as you want, but if you do, only one of those **classes can** contain concrete (implemented) methods. ... A **class can extend** at most one **abstract class**, but may implement many interfaces. That is, Java supports a limited form of **multiple inheritance**. May 24, 2011

[In Java, what is the difference between an abstract class and an ...](#)

<https://pythonconquerstheuniverse.wordpress.com/2011/05/24/java-abc-vs-interface/>

Search for: [Can a class extend multiple abstract classes?](#)

Can an abstract class extend a concrete class?

An **abstract class** always **extends a concrete class** (java.lang.Object at the very least). So it works the same as it always **does**. If you want to instantiate it, you **will** have to subclass it with a **concrete** implementation of those **abstract** methods and instantiate it through the **concrete class**. Just like you always do. Nov 13, 2015

[java - Abstract Class Extending Concrete Classes - Stack Overflow](#)

<https://stackoverflow.com/questions/.../abstract-class-extending-concrete-classes?rq=1>

Search for: [Can an abstract class extend a concrete class?](#)

Can static class be inherited?

Static classes are sealed and therefore cannot be **inherited**. They cannot **inherit** from any **class** except Object. **Static classes** cannot contain an instance constructor; however, they **can** have a **static** constructor. For more information, see **Static Constructors** (C# Programming Guide).

[c# - Why can't I inherit static classes? - Stack Overflow](#)

<https://stackoverflow.com/q/774181>

Search for: [Can static class be inherited?](#)

Can we declare abstract method in final class?

Yes, those **methods** cannot be overridden in subclasses. An example of that is the template **method** pattern... Of course, it means you **can** subclass it, but you cannot override that particular **method**. ... So **we can declare a method as final** in **Abstract class**, and it **will** be over ridden in subclass. Nov 12, 2011

[java - Can an abstract class have a final method? - Stack Overflow](#)

<https://stackoverflow.com/questions/1299398/can-an-abstract-class-have-a-final-method>

Search for: [Can we declare abstract method in final class?](#)

Can final class can be instantiated?

class, which cannot be **instantiated** to create its objects. **instantiated** if it overrides all abstract methods by implementation them. superclass abstract methods is itself abstract; and it cannot be **instantiated**. ¥ Java supports this is through the keyword **"final"**.

[Final and Abstract Classes Restricting Inheritance Abstract Classes ...](#)

www.buyya.com/254/Lectures/Lecture13.pdf

Search for: [Can final class can be instantiated?](#)

Can a subclass inherit from multiple superclasses?

Yes it is called as **Multiple Inheritance** where we **can inherit** the **subclass** from **two** or more than **two super classes**. This **can** be easily done in C++ but not in java. Since, in Java, we **can inherit** only one class at a time, it is not possible to perform **multiple inheritance**.

[Can a subclass inherit from more than one superclass in object ...](#)

<https://www.quora.com/Can-a-subclass-inherit-from-more-than-one-superclass-in-object...>

Search for: [Can a subclass inherit from multiple superclasses?](#)

Can subclass access private fields superclass?

Functionally, it's not inherited. But ideally, it is. A **subclass** does not inherit the **private** members of its parent class. However, if the **superclass** has public or protected methods for **accessing** its **private fields**, these **can** also be used by the **subclass**. Jan 18, 2011

[java - Do subclasses inherit private fields? - Stack Overflow](#)

<https://stackoverflow.com/questions/4716040/do-subclasses-inherit-private-fields>

Search for: [Can subclass access private fields superclass?](#)

Can we extend enum in Java?

All **enums** implicitly **extend** `java.lang.Enum`. In **Java**, a class **can** only **extend** one parent and therefore an **enum** cannot **extend** any other class (but implement interfaces – see below). **Extending Enum** means that every **enum** has a few methods that make it more usable: `static values()` Mar 27, 2017

[Fundamentals of Java Enum Types — SitePoint](#)

<https://www.sitepoint.com/fundamentals-of-java-enum-types-tutorial/>

Search for: [Can we extend enum in Java?](#)

Can we declare class as static?

But remember you **can** only **declare** a **static class** inside a top-level **class**, it is illegal to **declare** it inside an inner **class**. Nested **classes** (a **class** within a **class**) are the only ones that **can** be **declared static**. This is so the parent **class** **does** not have to be instantiated to access the nested **class**. Mar 16, 2013

[Why is class declared as static in Java? - Stack Overflow](#)

<https://stackoverflow.com/questions/15448352/why-is-class-declared-as-static-in-java>

Search for: [Can we declare class as static?](#)

Can enums be extended?

You **can** add more constants by making that new **enum** also **extend** the interface. Under the covers your **ENUM** is just a regular class generated by the compiler. That generated class **extends** `java.lang.Enum`. The technical reason you **can't extend** the generated class is that the generated class is **final**.

[java - Can enums be subclassed to add new elements? - Stack Overflow](#)

<https://stackoverflow.com/questions/.../can-enums-be-subclassed-to-add-new-elements>

Search for: [Can enums be extended?](#)

Can abstract class inherit object class?

Class C **does** not implement those methods, so compilation failure. **Object** is a superclass of **abstract classes**... but it's not a subclass, and subclasses are responsible for implementing **abstract** functions. ... In this case both A and C **inherit** the implementation of those methods from **Object** and no compilation error occurs. Jul 24, 2014

[java - Does abstract class extend Object? - Stack Overflow](#)

<https://stackoverflow.com/questions/24938564/does-abstract-class-extend-object>

Search for: [Can abstract class inherit object class?](#)

Can a class extend an interface?

An abstract **class** is still a **class**, just like any other. Only **interfaces** **can extend** other **interfaces**. You **can** actually **extend interfaces** in Java, but it would still be called an **interface**. Then you **can** use this extended **interface** implemented in your abstract **class**. Aug 9, 2014

[java - Why can't an abstract class extend an interface? - Stack ...](#)

<https://stackoverflow.com/questions/.../why-cant-an-abstract-class-extend-an-interface>

Search for: [Can a class extend an interface?](#)

Why Multiple inheritance is not supported in Java?

In **java** this can never occur as there is no **multiple inheritance**. Here even if two interfaces are going to have same method, the implementing class will have only one method and that too will be done by the implementer. Dynamic loading of classes makes the implementation of **multiple inheritance** difficult. Sep 24, 2012

[Why Multiple Inheritance is Not Supported in Java - Java Tutorial Blog](#)

<https://javapapers.com/core-java/why-multiple-inheritance-is-not-supported-in-java/>

Search for: [Why Multiple inheritance is not supported in Java?](#)

Can we create abstract and final class in Java?

This is the same reason why **abstract** methods cannot have access modifier private. you cannot inherit **final class**, but **abstract class** can be inherit into subclasses. A method **can** never, ever, ever be marked as both **abstract and final**, or both **abstract** and private. Oct 1, 2013

[c# - Why can't a Java class be both abstract and final - Stack ...](#)

<https://stackoverflow.com/questions/.../why-cant-a-java-class-be-both-abstract-and-final>

Search for: [Can we create abstract and final class in Java?](#)

Can an interface extends an abstract class?

Abstract classes can implement one or more **interfaces** and **can extend** one **abstract class** at most. ... **Abstract classes** can **extend** other at most one **abstract** or concrete **class** and implement several **interfaces**. Any **class** that does not implement all the **abstract** methods of its super **class** has to be an **abstract class** itself. Jul 18, 2017

[Interface vs Abstract class vs Concrete class – Heuristics – Medium](#)

<https://medium.com/heuristics/interface-vs-abstract-class-vs-concrete-class-196f20c3af9a>

Search for: [Can an interface extends an abstract class?](#)

Can we declare a class as abstract without having any abstract method?

Yes **we can** have an **abstract class without Abstract Methods** as both are independent concepts. **Declaring a class abstract** means that it **can** not be instantiated on its own and **can** only be sub classed. **Declaring a method abstract** means that **Method will** be defined in **the** subclass. Jan 27, 2011

[java - Defining an abstract class without any abstract methods ...](#)

<https://stackoverflow.com/.../defining-an-abstract-class-without-any-abstract-methods>

Search for: [Can we declare a class as abstract without having any abstract method?](#)

Can abstract class extend abstract class?

One thing to notice is that the second **abstract class** doesn't need to implement **abstract** methods from first **class**, but the first concrete must implement both. But it makes sense only if the **abstract** subclass adds more functionality (**abstract** or not). Yes you **can** do it. Jul 19, 2011

[java - Can one abstract class extend another abstract class and ...](#)

<https://stackoverflow.com/.../can-one-abstract-class-extend-another-abstract-class-and-in...>

Search for: [Can abstract class extend abstract class?](#)

Can an interface extend a class?

A Java **class** **can** only **extend** one parent **class**. Multiple inheritance is not allowed. **Interfaces** are not **classes**, however, and an **interface** **can extend** more than one parent **interface**. The **extends** keyword is used once, and the parent **interfaces** are declared in a comma-separated list.

[Java Interfaces - Tutorialspoint](#)

https://www.tutorialspoint.com/java/java_interfaces.htm

Search for: [Can an interface extend a class?](#)

Can an interface implement an abstract class?

In Java, an **abstract class can implement an interface**, and not provide implementations of all of the **interface's** methods. It is the responsibility of the first concrete **class** that has that **abstract class** as an ancestor to **implement** all of the methods in the **interface**.

[Why does an abstract class need to implement interface methods ...](#)

<https://hackerbits.com/.../why-does-an-abstract-class-need-to-implement-interface-metho...>

Search for: [Can an interface implement an abstract class?](#)

Why static class Cannot be inherited in Java?

Static methods are **inherited in Java** but they don't take part in polymorphism. If we attempt to override the **static** methods they will just hide the superclass **static** methods instead of overriding them. **Static** method is **inherited** in subclass but it is not polymorphism. Apr 24, 2012

[inheritance - Are static methods inherited in Java? - Stack Overflow](#)

<https://stackoverflow.com/questions/10291949/are-static-methods-inherited-in-java>

Search for: [Why static class Cannot be inherited in Java?](#)

How can you prevent your class from being subclassed?

1. Use final.
2. Use private constructors.
3. Use a comment: // do not inherit.
4. Use a javadoc comment.
5. Make every method final, so people can't override them.
6. Use a runtime check in the class constructor:

```
if (this.getClass() != MyClass.class) { throw new RuntimeException("Subclasses not allowed"); }
```

[java - Stopping inheritance without using final - Stack Overflow](#)

<https://stackoverflow.com/questions/451182/stopping-inheritance-without-using-final>

Search for: [How can you prevent your class from being subclassed?](#)

Can abstract class be inherited?

Abstract classes cannot be instantiated, but they **can** be subclassed. When an **abstract class** is subclassed, the subclass usually provides implementations for all of the **abstract** methods in its parent **class**. However, if it **does** not, then the subclass must also be declared **abstract**.

[Abstract Methods and Classes \(The Java™ Tutorials > Learning the ...](#)

<https://docs.oracle.com/javase/tutorial/java/landl/abstract.html>

Search for: [Can abstract class be inherited?](#)

Does abstract class have constructor?

Yes, when we define a **class** to be an **Abstract Class** it cannot be instantiated but that **does** not mean an **Abstract class** cannot **have a constructor**. Each **abstract class** must **have** a concrete subclass which will implement the **abstract** methods of that **abstract class**. Nov 4, 2008

[java - Can an abstract class have a constructor? - Stack Overflow](#)

<https://stackoverflow.com/questions/260666/can-an-abstract-class-have-a-constructor>

Search for: [Does abstract class have constructor?](#)

Can we declare interface as final?

A **final method** can't be overridden. That defies the purpose of having an **interface** if you cannot actually implement the **method**. All variables are implicitly public static and **final** in **interfaces**. Prior to Java 8, you **can't** create static **methods** in **interfaces**. May 20, 2012

[java - Why Interface methods cannot be "static" & "final"? - Stack ...](#)

<https://stackoverflow.com/questions/.../why-interface-methods-cannot-be-static-final>

Search for: [Can we declare interface as final?](#)

Can we declare abstract class as static?

However, for the same reason, you **can't declare a static** method to be **abstract**. Normally, the compiler **can** guarantee that an **abstract** method **will** have a real implementation any time that it is called, because you **can't** create an instance of an **abstract class**. Jul 1, 2013

[java - Can we use static method in an abstract class? - Stack Overflow](#)
<https://stackoverflow.com/questions/.../can-we-use-static-method-in-an-abstract-class>

Search for: [Can we declare abstract class as static?](#)

Can we inherit a final class?

Final Classes and Methods. **Inheritance** is surely one of the highly useful features in Java. ... A **class** declared as **final** cannot be extended while a method declared as **final** cannot be overridden in its subclasses. A method or a **class** is declared to be **final** using the **final** keyword.

[Final Classes and Methods - Java Tutorial - Java With Us](#)
www.javawithus.com/tutorial/final-classes-and-methods

Search for: [Can we inherit a final class?](#)

Can we extend multiple classes in Java?

Can a class inherit from multiple classes Java?

Does a subclass inherit both member variables and methods?

A **subclass does** not **inherit** the private **members** of its parent class. However, if the superclass has public or protected **methods** for accessing its private fields, these can also be used by the **subclass**. A nested class has access to all the private **members** of its enclosing class—**both** fields and **methods**.

[Inheritance \(The Java™ Tutorials > Learning the Java Language ...](#)
<https://docs.oracle.com/javase/tutorial/java/landl/subclasses.html>

Search for: [Does a subclass inherit both member variables and methods?](#)

Are private variables inherited Java?

Only the public and protected member are **inherited**. A subclass does not **inherit** the **private members** of its parent class. However, if the superclass has public or protected methods for accessing its **private** fields, these can also be used by the subclass. Jul 1, 2011

[Private members in Java inheritance - Stack Overflow](#)
<https://stackoverflow.com/questions/6543328/private-members-in-java.../10355759>

Search for: [Are private variables inherited Java?](#)

What is superclass and subclass in Java?

The derived class (the class that is derived from another class) is called a **subclass**. The class from which it's derived is called the **superclass**. ... In fact, in **Java**, all classes must be derived from some class.

[Subclasses, Superclasses, and Inheritance](#)
journals.ecs.soton.ac.uk/java/tutorial/java/javaOO/subclasses.html

Search for: [What is superclass and subclass in Java?](#)

Can we inherit private class in Java?

Only protected and public methods/variables **can** be **inherited** and/or overridden. **Private** means you **can** Only access it in that **class** and nowhere else. Subclasses **can** only invoke or override protected or public methods (or methods without access modifiers, if the superclass is in the same package) from their superclasses. Oct 19, 2013

[private method in inheritance in Java - Stack Overflow](#)
<https://stackoverflow.com/questions/19461867/private-method-in-inheritance-in-java>

Search for: [Can we inherit private class in Java?](#)

Can enum extend class in Java?

Answer: **Enum can** not **extend** any **class in java**, the reason is by default, **Enum extends** abstract base **class java.lang.Enum**. Since **java does** not support multiple inheritance for **classes**, **Enum can** not **extend** another **class**.

[Can Enum extend any class in Java? - Java Interview Questions ...](#)
www.java2novice.com/java_interview_questions/can-enum-extend/

Search for: [Can enum extend class in Java?](#)

Can enum implement interface?

The Comparable example given by several people here is wrong, since **Enum** already **implements** that. You **can't** even override it. Since **Enums can implement interfaces** they **can** be used for strict enforcing of the singleton pattern. Trying to make a standard class a singleton allows... Apr 26, 2010

[java - Why would an Enum implement an Interface? - Stack Overflow](#)
<https://stackoverflow.com/questions/.../why-would-an-enum-implement-an-interface>

Search for: [Can enum implement interface?](#)

Can enum be abstract?

An **enum** constant defines an instance of the **enum** type". An **enum** type is a subclass of **java.Lang.Enum**. An **enum** cannot extend anything else, but still it **can** implement other interfaces. ... You **can** find an **abstract** method **getCurrency()** inside **enum**. Nov 14, 2012

[Java Enum and Abstract Methods | 11th Hour](#)
<https://deepeshdarshan.wordpress.com/2012/11/14/java-enum-and-abstract-methods/>

Search for: [Can enum be abstract?](#)

Can a class be final?

You **can** declare some or all of a **class's** methods **final**. You use the **final** keyword in a method declaration to indicate that the method cannot be overridden by subclasses. ... A **class** that is declared **final** cannot be subclassed. This is particularly useful, for example, when creating an immutable **class** like the **String class**.

[Writing Final Classes and Methods \(The Java™ Tutorials > Learning ...](#)
<https://docs.oracle.com/javase/tutorial/java/landl/final.html>

Search for: [Can a class be final?](#)

Can static class have constructor?

Static constructors have the following properties: A **static constructor** does not take access modifiers or **have** parameters. A **static constructor** is called automatically to initialize the **class** before the first instance is created or any **static** members are referenced. A **static constructor** cannot be called directly. Jul 19, 2015

[Static Constructors \(C# Programming Guide\) | Microsoft Docs](#)
<https://docs.microsoft.com/en-us/dotnet/csharp/programming.../static-constructors>

Search for: [Can static class have constructor?](#)

Can static class have non static methods in Java?

A **static class** may however itself **contain non-static methods**. There's no such thing as a "**static class**" in **Java** (at least not in the way you seem to understand it). In **C#** there is a "**static class**" that **can** not be instantiated and **can't have non-static** members. Sep 11, 2012

[Does static class contain only static method in java? - Stack Overflow](#)
<https://stackoverflow.com/questions/.../does-static-class-contain-only-static-method-in-ja...>

Search for: [Can static class have non static methods in Java?](#)

Can enums be extended?
Can we extend enum in Java?
<p>Can we declare class as static?</p> <p>But remember you can only declare a static class inside a top-level class, it is illegal to declare it inside an inner class. Nested classes (a class within a class) are the only ones that can be declared static. This is so the parent class does not have to be instantiated to access the nested class. Mar 16, 2013</p> <p>Why is class declared as static in Java? - Stack Overflow https://stackoverflow.com/questions/15448352/why-is-class-declared-as-static-in-java</p> <p>Search for: Can we declare class as static?</p>
Can we create abstract and final class in Java?
<p>Can we declare abstract method in final class?</p> <p>Yes, those methods cannot be overridden in subclasses. An example of that is the template method pattern... Of course, it means you can subclass it, but you cannot override that particular method. ... So we can declare a method as final in Abstract class, and it will be over ridden in subclass. Nov 12, 2011</p> <p>java - Can an abstract class have a final method? - Stack Overflow https://stackoverflow.com/questions/1299398/can-an-abstract-class-have-a-final-method</p> <p>Search for: Can we declare abstract method in final class?</p>
Can abstract class inherit object class?
Can class extend interface?
Can interface extends abstract class?
Why Multiple inheritance is not supported in Java?
Why Java does not support multiple inheritance by means of classes?
<p>Can we override static method?</p> <p>Static methods can not be overridden because they are not part of the object's state. Rather, they belongs to the class (i.e they are class methods). It is ok to overload static (and final) methods. No, Static methods can't be overridden as it is part of a class rather than an object. Mar 26, 2011</p> <p>Can I override and overload static methods in Java? - Stack Overflow https://stackoverflow.com/questions/.../can-i-override-and-overload-static-methods-in-ja...</p> <p>Search for: Can we override static method?</p>
Why Multiple inheritance is supported by interfaces in Java?
<p>Can abstract class have constructor?</p> <p>Yes, when we define a class to be an Abstract Class it cannot be instantiated but that does not mean an Abstract class cannot have a constructor. Each abstract class must have a concrete subclass which will implement the abstract methods of that abstract class. Nov 4, 2008</p> <p>java - Can an abstract class have a constructor? - Stack Overflow https://stackoverflow.com/questions/260666/can-an-abstract-class-have-a-constructor</p> <p>Search for: Can abstract class have constructor?</p>
<p>Can we declare interface as final?</p> <p>A final method can't be overridden. That defies the purpose of having an interface if you cannot actually implement the method. All variables are implicitly public static and final in interfaces. Prior to Java 8, you can't create static methods in interfaces. May 20, 2012</p> <p>java - Why Interface methods cannot be "static" & "final"? - Stack ... https://stackoverflow.com/questions/.../why-interface-methods-cannot-be-static-final</p> <p>Search for: Can we declare interface as final?</p>

Can we make abstract class as final in Java?

Since you cannot **make** a **final class abstract** in **Java**, this becomes illegal and compiler prohibits you from doing that by throwing an error. But, yes, you **can** declare static methods on both **final class** as well as **abstract classes**, there is no problem with that. Jul 15, 2017

[Is it possible to have an abstract method in a final class? - Javarevisited](https://javarevisited.blogspot.com/.../is-it-possible-to-have-abstract-method-in-final-class...)
<https://javarevisited.blogspot.com/.../is-it-possible-to-have-abstract-method-in-final-class...>

Search for: [Can we make abstract class as final in Java?](#)

Can an interface extend a class?

Can a class extend multiple classes?

Can an abstract class extend a concrete class?

An **abstract class** always **extends a concrete class** (java.lang.Object at the very least). So it works the same as it always **does**. If you want to instantiate it, you **will** have to subclass it with a **concrete** implementation of those **abstract** methods and instantiate it through the **concrete class**. Just like you always do. Nov 13, 2015

[java - Abstract Class Extending Concrete Classes - Stack Overflow](https://stackoverflow.com/questions/.../abstract-class-extending-concrete-classes?rq=1)
<https://stackoverflow.com/questions/.../abstract-class-extending-concrete-classes?rq=1>

Search for: [Can an abstract class extend a concrete class?](#)

Can abstract class have concrete methods?

abstract classes can't be instantiated, only subclassed. other **classes** extend **abstract classes**. **can have** both **abstract** and **concrete methods**. similar to interfaces, but (1) **can** implement **methods**, (2) fields **can have** various access modifiers, and (3) subclasses **can** only extend one **abstract class**. Aug 10, 2018

[Java: Abstract classes and abstract methods | I'd Rather Be Writing](https://idratherbewriting.com/java-abstract-methods/)
<https://idratherbewriting.com/java-abstract-methods/>

Search for: [Can abstract class have concrete methods?](#)

Can abstract classes be instantiated?

Abstract classes cannot be **instantiated**, but they **can** be subclassed. When an **abstract class** is subclassed, the subclass usually provides implementations for all of the **abstract** methods in its parent **class**. However, if it does not, then the subclass must also be declared **abstract** .

[Abstract Methods and Classes \(The Java™ Tutorials > Learning the ...](https://docs.oracle.com/javase/tutorial/java/landl/abstract.html)
<https://docs.oracle.com/javase/tutorial/java/landl/abstract.html>

Search for: [Can abstract classes be instantiated?](#)

Can an abstract class have no abstract methods?

Yes we **can have** an **abstract class without Abstract Methods** as both are independent concepts. Declaring a **class abstract** means that it **can not** be instantiated on its own and **can** only be sub classed. Declaring a **method abstract** means that **Method will** be defined in the subclass. Jan 27, 2011

[java - Defining an abstract class without any abstract methods ...](https://stackoverflow.com/.../defining-an-abstract-class-without-any-abstract-methods)
<https://stackoverflow.com/.../defining-an-abstract-class-without-any-abstract-methods>

Search for: [Can an abstract class have no abstract methods?](#)

Why abstract class can not be instantiated?

What is difference between interface and abstract class in Java?

1.Main **difference** is methods of a **Java interface** are implicitly **abstract** and cannot have implementations. A **Java abstract class** can have instance methods that implements a default behavior. 2.Variables declared in a **Java interface** is by default final. An **abstract class** may contain non-final variables. Dec 16, 2009

[oop - What is the difference between an interface and abstract ...](https://stackoverflow.com/.../what-is-the-difference-between-an-interface-and-abstract-cl...)
<https://stackoverflow.com/.../what-is-the-difference-between-an-interface-and-abstract-cl...>

Search for: [What is difference between interface and abstract class in Java?](#)

Can an abstract class extend another abstract class?

One thing to notice is that the second **abstract class** doesn't need to implement **abstract** methods from first **class**, but the first concrete must implement both. But it makes sense only if the **abstract** subclass adds more functionality (**abstract** or not). Yes you **can** do it. Jul 19, 2011

[java - Can one abstract class extend another abstract class and ...](#)
<https://stackoverflow.com/.../can-one-abstract-class-extend-another-abstract-class-and-in...>

Search for: [Can an abstract class extend another abstract class?](#)

Can an interface implement an abstract class?

In Java, an **abstract class can implement an interface**, and not provide implementations of all of the **interface's** methods. It is the responsibility of the first concrete **class** that has that **abstract class** as an ancestor to **implement** all of the methods in the **interface**.

[Why does an abstract class need to implement interface methods ...](#)
<https://hackerbits.com/.../why-does-an-abstract-class-need-to-implement-interface-metho...>

Search for: [Can an interface implement an abstract class?](#)

Can class inherit interface?

Can an interface implement a class?

implements denotes defining an **implementation** for the methods of an **interface**. However **interfaces** have no **implementation** so that's not possible. However, **interface** is an 100% abstract **class**, and an abstract **class can implement interfaces** (100% abstract **class**) without **implement** its methods. Oct 13, 2010

[java - Why an interface can not implement another interface ...](#)
<https://stackoverflow.com/.../why-an-interface-can-not-implement-another-interface>

Search for: [Can an interface implement a class?](#)

Can abstract class be implemented?

Simply because, just like an **abstract class**, an interface **can NOT** be instantiated. So, it is not required to explicitly mention the methods from its "parent". However, ALL the parent method signatures DO implicitly become a part of the extending interface or **implementing abstract class**. Oct 13, 2008

[java - Why an abstract class implementing an interface can miss ...](#)
<https://stackoverflow.com/.../why-an-abstract-class-implementing-an-interface-can-miss-t...>

Search for: [Can abstract class be implemented?](#)

Can abstract class have static methods in Java?

However, for the same reason, you **can't** declare a **static method** to be **abstract**. Normally, the compiler **can** guarantee that an **abstract method** will **have** a real implementation any time that it is called, because you **can't** create an instance of an **abstract class**. Jul 1, 2013

[java - Can we use static method in an abstract class? - Stack Overflow](#)
<https://stackoverflow.com/questions/.../can-we-use-static-method-in-an-abstract-class>

Search for: [Can abstract class have static methods in Java?](#)

Can an interface implement methods?

Can you extend a static class in Java?

Can static class inherit another class?

Can static method be final?

How can you prevent your class from being subclassed?

Can we inherit main method in Java?

The **inheritance** is working as usual here and the public static void **main()** is **inherited** by Factorial which is why you see your output on executing **java** Factorial . You **can** have more than **one** class in the same file, but only **one** public , as Base isn't a public class, but it's not a recommended practice. Jun 6, 2013

[Java Extending class containing main method - Stack Overflow](https://stackoverflow.com/questions/.../java-extending-class-containing-main-method)
<https://stackoverflow.com/questions/.../java-extending-class-containing-main-method>

Search for: [Can we inherit main method in Java?](#)

How do you prevent a method from being overridden?

Can we override static method in Java?

Can we inherit a final class?

What is a abstract class in Java?

Can we declare abstract method in final class?

Can we create abstract and final class in Java?

Can abstract class have constructor?

Yes, when we define a **class** to be an **Abstract Class** it cannot be instantiated but that does not mean an **Abstract class** cannot **have a constructor**. Each **abstract class** must **have** a concrete subclass which will implement the **abstract** methods of that **abstract class**. Nov 4, 2008

[java - Can an abstract class have a constructor? - Stack Overflow](https://stackoverflow.com/questions/260666/can-an-abstract-class-have-a-constructor)
<https://stackoverflow.com/questions/260666/can-an-abstract-class-have-a-constructor>

Search for: [Can abstract class have constructor?](#)

Can interface class have constructor?

Can abstract class have data attributes?

An **abstract** method **has** no implementation. Typically an **abstract class** contains one or more **abstract** method. The diagram also shows three subclasses that inherit behaviour and **data attributes** from the Employee **class**. These are concrete **classes** that **can** be instantiated; **abstract classes** cannot directly be instantiated.

[Abstract and Concrete Classes - CEMS](http://www.cems.uwe.ac.uk/~jsa/UMLJavaShortCourse09/CGOutput/Unit9/.../page_03.htm)
www.cems.uwe.ac.uk/~jsa/UMLJavaShortCourse09/CGOutput/Unit9/.../page_03.htm

Search for: [Can abstract class have data attributes?](#)

Can abstract class have main method?

Why an interface Cannot be final?

Can we declare interface as static?

If you want statics, use an abstract **class** and inherit it, otherwise, remove the **static**. You **can't define static methods** in an **interface** because **static methods** belongs to a **class** not to an instance of **class**, and **interfaces** are not Classes. Feb 5, 2009

[Why can't I define a static method in a Java interface? - Stack ...](https://stackoverflow.com/.../why-cant-i-define-a-static-method-in-a-java-interface)
<https://stackoverflow.com/.../why-cant-i-define-a-static-method-in-a-java-interface>

Search for: [Can we declare interface as static?](#)

Can we declare interface as private?

Can abstract class be instantiated?

Can abstract class have static and final variables?

What's the difference between abstract class and interface?

Can we extend final class in Java?

You **can't extend** a **class** that is declared **final** . However you **can** create an intermediate **class**, called Wrapper . This Wrapper **will** basically contain an instance of the original **class** and provide alternative methods to modify the state of the object according what you want to **do**. Nov 9, 2012

[static - How to extend a final class in Java - Stack Overflow](https://stackoverflow.com/questions/13294629/how-to-extend-a-final-class-in-java)

<https://stackoverflow.com/questions/13294629/how-to-extend-a-final-class-in-java>

Search for: [Can we extend final class in Java?](#)

Why can't we declare a constructor as final?

When you set a method as **final** it means: "You don't want any class override it." But the **constructor** (according to the Java Language Specification) can't be overridden, so it is clean. ... When you set a method as static it means: "The method belongs to the class, not a particular object." Feb 28, 2012

[java - Why constructors cannot be final - Stack Overflow](https://stackoverflow.com/questions/9477476/why-constructors-cannot-be-final)

<https://stackoverflow.com/questions/9477476/why-constructors-cannot-be-final>

Search for: [Why can't we declare a constructor as final?](#)

Is String a final class?

String is a very core **class** in Java, many things rely on it working a certain way, for example being immutable. Making the **class final** prevents subclasses that could break these assumptions. Note that, even now, if you use reflection, you can break **Strings** (change their value or hashCode). Jan 15, 2010

[Why is String class declared final in Java? - Stack Overflow](https://stackoverflow.com/questions/2068804/why-is-string-class-declared-final-in-java)

<https://stackoverflow.com/questions/2068804/why-is-string-class-declared-final-in-java>

Search for: [Is String a final class?](#)

Can a class extend multiple classes?

Does a subclass inherit both member variables and methods?

A **subclass** does not **inherit** the private **members** of its parent class. However, if the superclass has public or protected **methods** for accessing its private fields, these can also be used by the **subclass**. A nested class has access to all the private **members** of its enclosing class—**both** fields and **methods**.

[Inheritance \(The Java™ Tutorials > Learning the Java Language ...](https://docs.oracle.com/javase/tutorial/java/landl/subclasses.html)

<https://docs.oracle.com/javase/tutorial/java/landl/subclasses.html>

Search for: [Does a subclass inherit both member variables and methods?](#)

Do subclasses inherit constructors?

The **constructor** is not **inherited** but you are still able to call it. No a **subclass** cannot **inherit** the **constructors** of its superclass. **Constructors** are special function members of a class in that they are not **inherited** by the **subclass**. **Constructors** are used to give a valid state for an object at creation. Mar 4, 2011

[java - Does a subclass inherit constructors from its super class ...](https://stackoverflow.com/.../does-a-subclass-inherit-constructors-from-it-super-class)

<https://stackoverflow.com/.../does-a-subclass-inherit-constructors-from-it-super-class>

Search for: [Do subclasses inherit constructors?](#)

Are private variables inherited in Java?

Only the public and protected member are **inherited**. A subclass does not **inherit** the **private members** of its parent class. However, if the superclass has public or protected methods for accessing its **private** fields, these can also be used by the subclass. Jul 1, 2011

[Private members in Java inheritance - Stack Overflow](https://stackoverflow.com/questions/6543328/private-members-in-java.../10355759)

<https://stackoverflow.com/questions/6543328/private-members-in-java.../10355759>

Search for: [Are private variables inherited in Java?](#)

Can we have a private class in Java?

private classes are allowed but only as inner or nested **classes**. If you have a **private** inner or nested **class**, then access is restricted to the scope of that outer **class**. ... We **can** not declare outer **class** as **private**. More precisely we **can** not use **private** access specifier with outer **class**. Dec 16, 2009

[Java: Why can we define a top level class as private? - Stack Overflow](#)

<https://stackoverflow.com/questions/.../java-why-can-we-define-a-top-level-class-as-priv...>

Search for: [Can we have a private class in Java?](#)

Can we inherit private class in Java?

What is superclass and subclass in Java?

Which class is superclass of every class in Java?

Can a subclass inherit from multiple superclasses?

Can we inherit final method in Java?

Can we overload final method in Java?

Can private class be inherited in Java?

Can a class extend enum in Java?

Answer: **Enum can not extend** any **class** in **java**, the reason is by default, **Enum extends** abstract base **class java.lang.Enum**. Since **java does** not support multiple inheritance for **classes**, **Enum can not extend** another **class**.

[Can Enum extend any class in Java? - Java Interview Questions ...](#)

www.java2novice.com/java_interview_questions/can-enum-extend/

Search for: [Can a class extend enum in Java?](#)

Can we extend enum class in Java?

Can we declare class as static?

But remember you **can** only **declare** a **static class** inside a top-level **class**, it is illegal to **declare** it inside an inner **class**. Nested **classes** (a **class** within a **class**) are the only ones that **can** be **declared static**. This is so the parent **class does** not have to be instantiated to access the nested **class**. Mar 16, 2013

[Why is class declared as static in Java? - Stack Overflow](#)

<https://stackoverflow.com/questions/15448352/why-is-class-declared-as-static-in-java>

Search for: [Can we declare class as static?](#)

Can enum implement interface?

The Comparable example given by several people here is wrong, since **Enum** already **implements** that. You **can't** even override it. Since **Enums can implement interfaces** they **can** be used for strict enforcing of the singleton pattern. Trying to make a standard class a singleton allows... Apr 26, 2010

[java - Why would an Enum implement an Interface? - Stack Overflow](#)

<https://stackoverflow.com/questions/.../why-would-an-enum-implement-an-interface>

Search for: [Can enum implement interface?](#)

Can enums be extended?

You **can** add more constants by making that new **enum** also **extend** the interface. Under the covers your **ENUM** is just a regular class generated by the compiler. That generated class **extends** **java.lang.Enum**. The technical reason you **can't extend** the generated class is that the generated class is **final**.

[java - Can enums be subclassed to add new elements? - Stack Overflow](#)

<https://stackoverflow.com/questions/.../can-enums-be-subclassed-to-add-new-elements>

Search for: [Can enums be extended?](#)

Can Java interface have static methods?

Java interface static method helps us in providing security by not allowing implementation classes to override them. ... This is because it's not allowed in **java**, since **Object** is the base class for all the classes and we **can't have** one class level **static method** and another instance **method** with same signature. Apr 2, 2018

[Java 8 Interface Changes - static method, default method - JournalDev](#)

<https://www.journaldev.com/2752/java-8-interface-changes-static-method-default-method>

Search for: [Can Java interface have static methods?](#)

Can we have abstract method in enum?

Yes, you **can** define **abstract methods** in an **enum** declaration if and only if all **enum** values **have** custom class bodies with implementations of those **methods** (i.e. no concrete **enum** value may be lacking an implementation). Sep 14, 2011

[java - Can an enum have abstract methods? - Stack Overflow](https://stackoverflow.com/questions/7413872/can-an-enum-have-abstract-methods)
<https://stackoverflow.com/questions/7413872/can-an-enum-have-abstract-methods>

Search for: [Can we have abstract method in enum?](#)

Can we override enum in Java?

Enums can have constant-specific method implementations which simply means they **can** have abstract methods and they **can override** it. Like classes in **Java**, **Enums can** implement interfaces. The important point is they **can** not extend from any other class because **Enums** implicitly extend from **java.lang.Enum**. Feb 23, 2015

[Java Enums Tutorial · Burak Aktas](http://buraktas.com/java-enums-tutorial/)
buraktas.com/java-enums-tutorial/

Search for: [Can we override enum in Java?](#)

Can we declare abstract method in final class?

Can we inherit final class in Java?

Final Classes and Methods. **Inheritance** is surely **one** of the highly useful features in **Java**. ... A **class** declared as **final** cannot be extended while a method declared as **final** cannot be overridden in its subclasses. A method or a **class** is declared to be **final** using the **final** keyword.

[Final Classes and Methods - Java Tutorial - Java With Us](http://www.javawithus.com/tutorial/final-classes-and-methods)
www.javawithus.com/tutorial/final-classes-and-methods

Search for: [Can we inherit final class in Java?](#)

Can we create abstract and final class in Java?

Can a static class have a constructor in Java?

Constructors are used to initialize a newly created object. In **Java**, **static** means the member (like a method or variable) is at the **class** level, not the object level - therefore a **static constructor** doesn't make sense. A **static** initializer is used to set up a **class**, if needed.

[Java \(programming language\): Why can't constructors be static? - Quora](https://www.quora.com/Java-programming-language-Why-cant-constructors-be-static)
<https://www.quora.com/Java-programming-language-Why-cant-constructors-be-static>

Search for: [Can a static class have a constructor in Java?](#)

Can we inherit static class?

Static classes are sealed and therefore cannot be **inherited**. They cannot **inherit** from any **class** except Object. **Static classes** cannot contain an instance constructor; however, they **can** have a **static** constructor. For more information, see **Static Constructors** (C# Programming Guide).

[c# - Why can't I inherit static classes? - Stack Overflow](https://stackoverflow.com/q/774181)
<https://stackoverflow.com/q/774181>

Search for: [Can we inherit static class?](#)

Can static class have constructor?

Static constructors have the following properties: A **static constructor** does not take access modifiers or **have** parameters. A **static constructor** is called automatically to initialize the **class** before the first instance is created or any **static** members are referenced. A **static constructor** cannot be called directly. Jul 19, 2015

[Static Constructors \(C# Programming Guide\) | Microsoft Docs](https://docs.microsoft.com/en-us/dotnet/csharp/programming.../static-constructors)
<https://docs.microsoft.com/en-us/dotnet/csharp/programming.../static-constructors>

Search for: [Can static class have constructor?](#)

Can a non static class have static methods Java?
<p>Can static class be instantiated Java?</p> <p>In Java, nested classes are not static by default (this non-static variety is also called an "inner class"), which means they require an instance of their outer class, which they track behind the scenes in a hidden field -- but this lets inner classes refer to fields of their associated enclosing class. Aug 18, 2013</p> <p>Can a static nested class be instantiated in Java? - Stack Overflow https://stackoverflow.com/questions/.../can-a-static-nested-class-be-instantiated-in-java</p> <p>Search for: Can static class be instantiated Java?</p>
Can we extend static class in Java?
Can a class be final?
Can we declare outer class as static?
Can we create object of abstract class?
Does abstract class have constructor?
Can we declare interface as final?
Can we declare abstract class as static?
Can we declare main method as final?
Can static method be final?
Why we Cannot override static method?
Can interface class have constructor?
Why abstract class can not be instantiated?
Can we have main method in abstract class?
Can we declare interface as static?
Can we declare interface as private?
Do interface methods have to be public?
What is the difference between abstract class and interface in Java?
Can an abstract class extend a concrete class?
Can we inherit abstract class in Java?
Can a class extend multiple classes?
Can abstract class extend abstract class?
Can class extend multiple abstract classes?
Can we declare abstract method in final class?
Can abstract class have concrete methods Java?
Can we create abstract and final class in Java?
Why Multiple inheritance is not supported in Java?
Why does an abstract class have a constructor?
Can a interface be instantiated?
Can an abstract method be defined in a non abstract class?
Can we create abstract method in non abstract class?
Can we call abstract method in Java?
Can abstract class have constructor in Java?
WHAT IS interface and abstract class in Java?
What is the difference between abstraction and encapsulation?
Can an abstract class extend an interface?
Can an interface implement another interface?