**What are the Rules for defining a constructor?constructor**

**What is the use of Private Constructors in Java?constructor**

**Can we have a Constructor in an Interface?constructor**

**What is Constructor Chaining in Java?constructor**

**Can a Constructor return any value ?constructor**

**Can an abstract class in Java have a constructor?constructor**

**Why constructors cannot be final in Java?constructor**

**Why constructors cannot be abstract in Java?constructor**

**Why constructors cannot be static in Java?constructor**

**What is Java JDK, JRE and JVM?jvm**

**when i will use only jdk and when jre, jvm?jvm**

**Explain public static void main(String args[]) in Java?core**

**Why Java is platform independent?core**

**what is static{} in java?core**

**what is {} in java?core**

**Execution order of static{}, {}, constructor?core**

**What is abstract method in Java?abstract**

An abstract method is a method without body. You just declare method, without defining it and use abstract keyword in method declaration. All method declared inside Java Interface are by default abstract. Here is an example of abstract method in Java

public void abstract printVersion();

abstract void sum(int a, int b);

Now, In order to implement this method, you need to extend abstract class and override this method.

**What is abstract class in Java?abstract**

An abstract class is sort of like a template, or an empty/partially empty structure, you have to extend it and build on it before you can use it.

**When Is An Abstract Method Used?abstract**

**From the Oracle tutorials :**

Unlike interfaces, abstract classes can contain fields that are not static and final, and they can contain implemented methods. Such abstract classes are similar to interfaces, except that they provide a partial implementation, leaving it to subclasses to complete the implementation. If an abstract class contains only abstract method declarations, it should be declared as an interface instead.

Multiple interfaces can be implemented by classes anywhere in the class hierarchy, whether or not they are related to one another in any way. Think of Comparable or Cloneable, for example.

By comparison, abstract classes are most commonly subclassed to share pieces of implementation. A single abstract class is subclassed by similar classes that have a lot in common (the implemented parts of the abstract class), but also have some differences (the abstract methods)

**Is it necessary for abstract class to have abstract method?abstract**

No, It’s not mandatory for an abstract class to have any abstract method. You can make a class abstract in Java, by just using abstract keyword. Compiler will enforce all structural restriction, applied to abstract class, e.g. now allowing to create any instance.

**How To Define An Abstract Class?abstract**

**Can you create instance of abstract class?abstract**

No, you can’t. You can however create an instance of the sub class of an abstract class to refer to it. You need to extend an abstract class and define it to make use of it.Java compiler will throw error, when a code tries to instantiate abstract class.

**Can we declare an interface with the abstract keyword?abstract**

Can We Declare Abstract Method In Non-abstract Class?abstract

**Can abstract class have constructors in Java?abstract**

Yes, abstract class can declare and define constructor in Java. Since you can not create instance of abstract class, constructor can only be called during constructor chaining, i.e. when you create instance of concrete implementation class.

**what is the purpose of constructor, if you can not instantiate abstract class?**

Well, it can still be used to initialize common variables, which are declared inside abstract class, and used by various implementation.

Also even if you don’t provide any constructor, compiler will add default no argument constructor in an abstract class, without that your subclass will not compile, since first statement in any constructor implicitly calls super(), default super class constructor in Java.

**Abstract Class Vs Interface?abstract**

**What Will Happen If We Do Not Override All The Abstract Methods In Sub-class?abstract**

**Does interface extend Object class by default?abstract**

**Can We Declare Abstract Method As Static?abstract**

Yes, abstract class can declare and define static methods, nothing prevents from doing that. But, you must follow guidelines for making a method static in Java, as it’s not welcomed in a object oriented design, because static methods can not be overridden in Java. It’s very rare, you see static methods inside abstract class, but as I said, if you have very good reason of doing it, then nothing stops you.

**Can We Declare Abstract Method As Private?abstract**

No. Abstract methods can not be private. If abstract methods are allowed to be private, then they will not be inherited to sub class and will not get enhanced.

**Can An Interface Be Extended By Another Interface In Java?abstract**

**Can abstract class implements interface in Java?abstract**

Yes, abstract class can implement interface by using implements keyword. Since they are abstract, they don’t need to implement all methods.

One example of this is java.util.List interface and corresponding java.util.AbstractList abstract class. Since AbstractList implements all common methods, concrete implementations like LinkedList and ArrayList are free from burden of implementing all methods, had they implemented List interface directly. It’s best of both world, you can get advantage of interface for declaring type, and flexibility of abstract class to implement common behavior at one place. Effective Java has a nice chapter on how to use interface and abstract class in Java, which is worth reading.

Do they require to implement all methods (but Test class needs to oveeride interface method)?abstract

**Can abstract class contains main method in Java?abstract**

Yes, abstract class can contain main method, it just another static method and you can execute Abstract class with main method, until you don’t create any instance

**Can abstract class be final in Java?abstract**

No, abstract class can not be final in Java. Making them final will stop abstract class from being extended, which is the only way to use abstract class. They are also opposite of each other, abstract keyword enforces to extend a class, for using it, on the other hand, final keyword prevents a class from being extended. In real world also, abstract signifies incompleteness, while final is used to demonstrate completeness. Bottom line is, you can not make your class abstract and final in Java, at same time, it’s a compile time error.

**When do you favor abstract class over interface**

Since it’s almost impossible to add a new method on a published interface, it’s better to use abstract class, when evolution is concern. Abstract class in Java evolves better than interface. Similarly, if you have too many methods inside interface, you are creating pain for all it’s implementation, consider providing an abstract class for default implementation. This is the pattern followed in Java collection package, you can see AbstractList provides default implementation for List interface.

What are different types of Inheritance supported by Java?inheritance

Why Java doesn't support multiple inheritance and why only through interface?inheritance

Can a class extend itself?inheritance

What happens if we write this() as a first statement in our constructor?inheritance

Can constructor have both super() & this()?inheritance

How do you prevent a field or a method of any class from inheriting to sub classes?inheritance

Can Constructors also be inherited to sub classes?inheritance

Can we override static method in Java?inheritance

Can we overload a static method in Java?inheritance

Can we override a private method in Java?inheritance

Can an interface extends more than one interface in Java?inheritance

What will happen if a class extends two interfaces and they both have a method with same name and signature?inheritance

What is Encapsulation in Java and Why is it called Data hiding?Encapsulation

How to achieve Data hiding programmatically?Encapsulation

What is the difference between Abstraction and Encapsulation?

What is polymorphism and what are the types of it?polymorphism

Method signature consists of?polymorphism

Can we override protected method as private?polymorphism

Does Java support operator overloading?polymorphism

Can we access super class version of overridden method in the sub class. If yes, how?polymorphism

Can we remove throws clause of a method while overriding it?polymorphism

What are the possible access modifiers a protected method can have if it is overridden in the sub class?polymorphism

Is String a keyword in java?string

What is string constant pool?,string

Why String is Immutable in Java?string