|  |  |
| --- | --- |
| Can abstract class have Constructor in Java | Yes, an abstract class can have a constructor in Java.  You can either explicitly provide a constructor to abstract class or if you don't, the compiler will add default constructor of no argument in abstract class.  This is true for all classes and it also applies to an abstract class.  For those who want to recall what is an abstract class in Java, it's a class which can not be instantiated with new() operator or any other ways.  In order to use an abstract class in Java, You need to extend it and provide a concrete class.  Abstract class is commonly used to **define a base class for a type hierarchy with default implementation**, which is applicable to all child classes. |
| Why can an abstract class have a constructor in Java? | Now if we say we can not create an instance of an abstract class then why do Java adds a constructor in the abstract class.  One of the reasons which make sense is when any class extends an abstract class, the constructor of sub class will invoke the constructor of super class either implicitly or explicitly.  This chaining of constructors is one of the reasons abstract class can have constructors to initialize class variables. |
| **public** **abstract** **void** sum();  **abstract** is Compulsory in Abstract class methods |  |
|  | The main difference between an abstract class and interface in Java 8 is the fact that an abstract class is a class and an interface is an interface.  A class can have a state which can be modified by non-abstract methods but an interface cannot have the state because they can't have instance variables.  The second difference is that an interface cannot have a constructor even in Java 8 but you may remember that abstract class always has a constructor in Java.  In reality, default or **defender methods are introduced to maintain backward compatibility** and same time making Collection API more suitable to be used inside key Java 8 features like lambda expressions.  Without adding default methods, it wasn't possible to declare any new method on existing interface in Java without breaking all classes which implement it, but because of default method, you can now better evolve your API. |
| Abstract Class |  |
| Interface |  |
|  | Java 9 interfaces still cannot contain constructors.  Java 9 interfaces still cannot have non-static members. |
|  |  |