**Why Multiple Inheritance is Not Supported in Java**

In an white paper titled “Java: an Overview” by James Gosling in February 1995 gives an idea on why multiple inheritance is not supported in Java.

*JAVA omits many rarely used, poorly understood, confusing features of C++ that in our experience bring more grief than beneﬁt. This primarily consists of operator overloading (although it does have method overloading),****multiple inheritance****, and extensive automatic coercions.*

Who better than Dr. James Gosling is qualified to make a comment on this. This paragraph gives us an overview and he touches this topic of not supporting multiple-inheritance.

**Java does not support multiple inheritance**

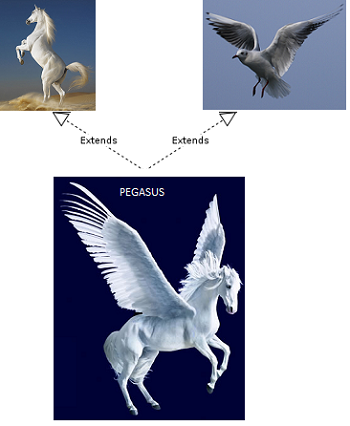
First lets nail this point. This itself is a point of discussion, whether java supports multiple inheritance or not. Some say, it supports using interface. No. There is no support for multiple inheritance in java. If you do not believe my words, read the above paragraph again and those are words of the father of Java.

This story of supporting multiple inheritance using interface is what we developers cooked up. Interface gives flexibility than concrete classes and we have option to implement multiple interface using single class. This is by agreement we are adhering to two blueprints to create a class.

This is trying to get closer to multiple inheritance. What we do is implement multiple interface, here we are not extending (inheriting) anything. The implementing class is the one that is going to add the properties and behavior. It is not getting the implementation free from the parent classes. I would simply say, there is no support for multiple inheritance in java.

**Multiple Inheritance**

Multiple inheritance is where we inherit the properties and behavior of multiple classes to a single class. C++, Common List, Scala are some popular languages that support multiple inheritance.



**Why Java does not support multiple inheritance?**

Now we are sure that there is no support for multiple inheritance in java. But why? This is a design decision taken by the creators of java. The keyword is simplicity and rare use.

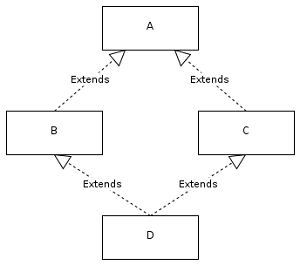
**Simplicity**

I want to share the definition for java given by James Gosling.

*JAVA: A simple, object oriented, distributed, interpreted, robust, secure, architecture neutral, portable, high performance, multithreaded, dynamic language.*

Look at the beauty of this definition for java. This should be the definition for a modern software language. What is the first characteristic in the language definition? It is simple.

In order to enforce simplicity should be the main reason for omitting multiple inheritance. For instance, we can consider**diamond problem of multiple inheritance**.



We have two classes B and C inheriting from A. Assume that B and C are [overriding](http://javapapers.com/core-java/overloading-and-overriding/) an inherited method and they provide their own implementation. Now D inherits from both B and C doing multiple inheritance. D should inherit that overridden method, which overridden method will be used? Will it be from B or C? Here we have an ambiguity.

In C++ there is a possibility to get into this trap though it provides alternates to solve this. 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.

**Rarely Used**

We have been using java for long now. How many times have we faced a situation where we are stranded and facing the wall because of the lack of support for multiple inheritance in java? With my personal experience I don’t remember even once. Since it is rarely required, multiple inheritance can be safely omitted considering the complexity it has for implementation. It is not worth the hassle and the path of simplicity is chosen.

Even if it is required it can be substituted with alternate design. So it is possible to live without multiple inheritance without any issues and that is also one reason.

My opinion on this is, omitting support for multiple inheritance in java is not a flaw and it is good for the implementers.