**Abstraction**

ABSTRACTION is a process of **hiding the implementation details and showing only functionality to the user**.

WAYS TO ACHIEVE ABSTRACTION

There are two ways to achieve abstraction in java

1.Abstract class (0 to 100%)

2.Interface (100%)

**Abstract Class**

abstract keyword

The 'abstract' keyword is a non-access modifier, used for classes and methods.

Abstract class

A CLASS which is declared with the 'abstract' keyword is known as an ABSTRACT CLASS in Java.

Rules for abstract classes

1. An Abstract class must be declared using 'abstract' keyword.

2. It can’t be instantiated (no object creation).

3. It can have Abstract, non-abstract, static, final methods.

4. It can have constructors.

Abstract method

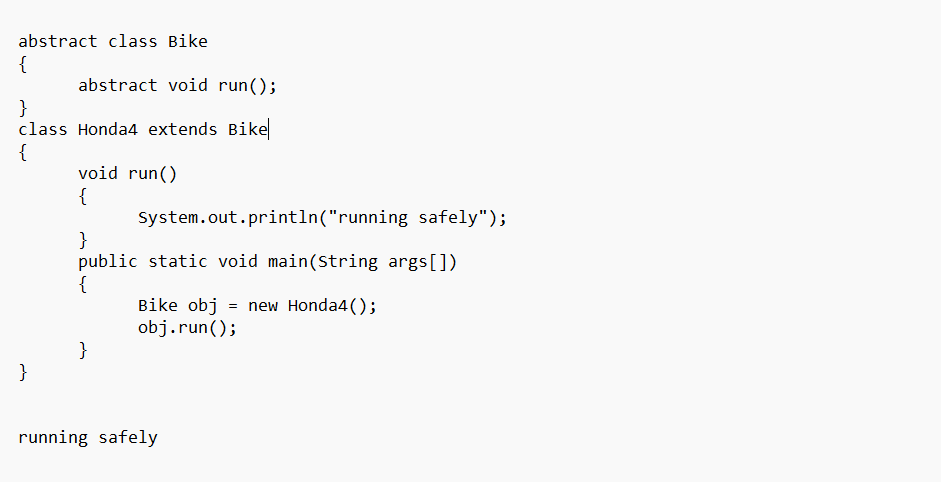
A method which is declared as 'abstract' and does not have body(implementation).

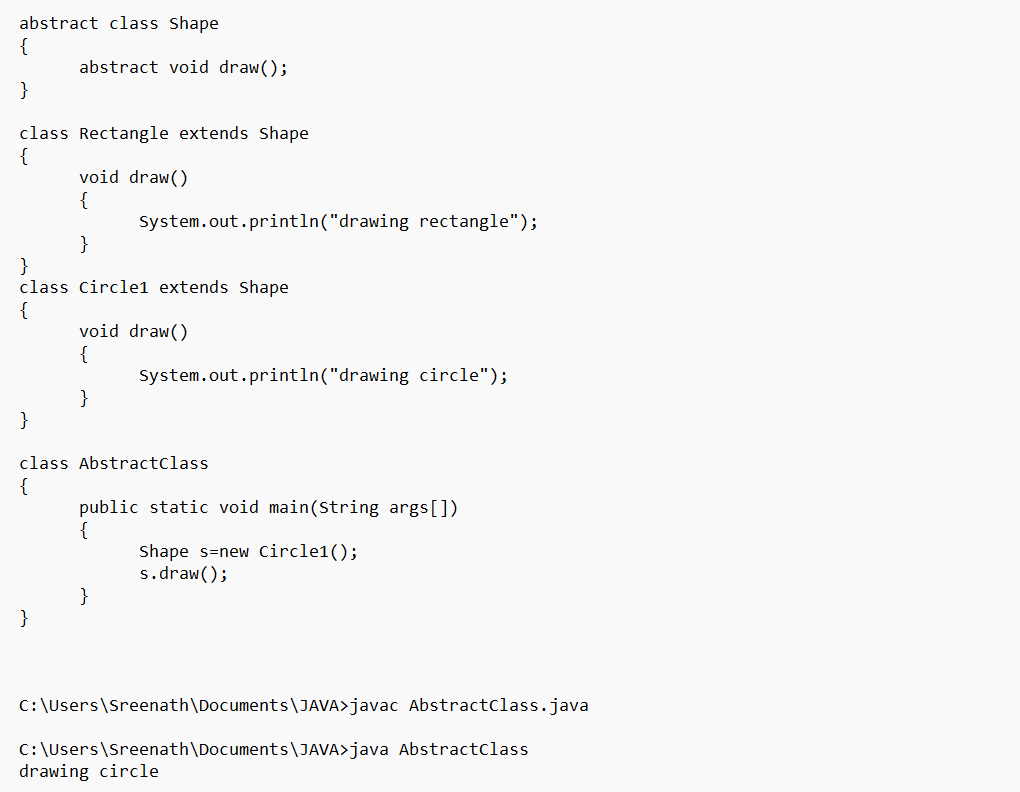
Rules for abstract method

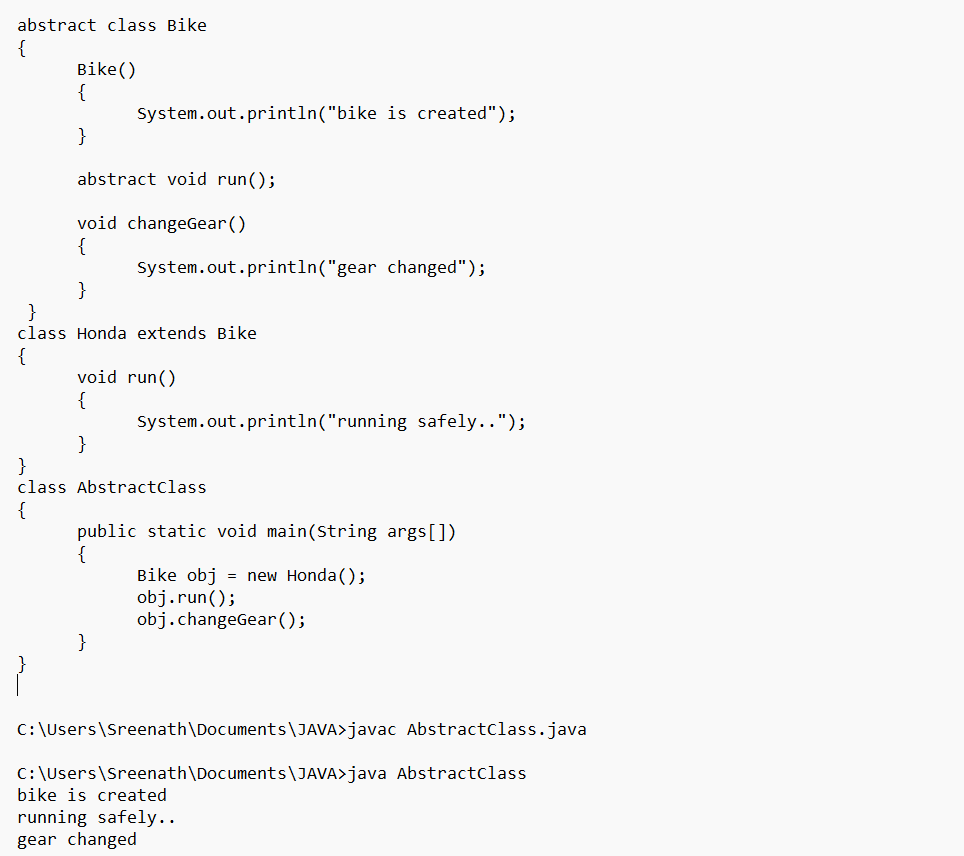
1. An abstract method must be declared in an Abstract class only.

2. If you are extending an abstract class that has an abstract method, you must either provide the implementation of the method or make this class abstract.

Examples







**Interfaces**

An Interface is an abstract type that is **used to specify the behaviour that all the classes must implement**.

Use of java Interface

1. abstraction

2. multiple inheritance

key points

1. INSTANTIATION

we can't instantiate (creating object) an interface.

2. Interface do not contain constructors.

3. ABSTRACTION

Interface provides full abstraction as none of its methods have body.

4. IMPLEMENTING AN INTERFACE

The keyword 'implements' is used by a class to implement an interface.

class must implement all the methods declared in an interface, or else it must be declared as an Abstract class.

5. Interface cannot be declared as private, protected. It is always public.

6. All the interface methods by default abstract and public.

7. Variables declared in an interface are by default public, static and final.

8. Interface variables must be initialized at the time of declaration otherwise compiler will throw an error.

9. we can have default and static methods in an interface. ( since java 8)

10. we can have private methods in an interface.

11. An interface can extend multiple interfaces.

Multiple Inheritance

If a class implements multiple interfaces, or an interface extends multiple interfaces, it is known as multiple inheritance.

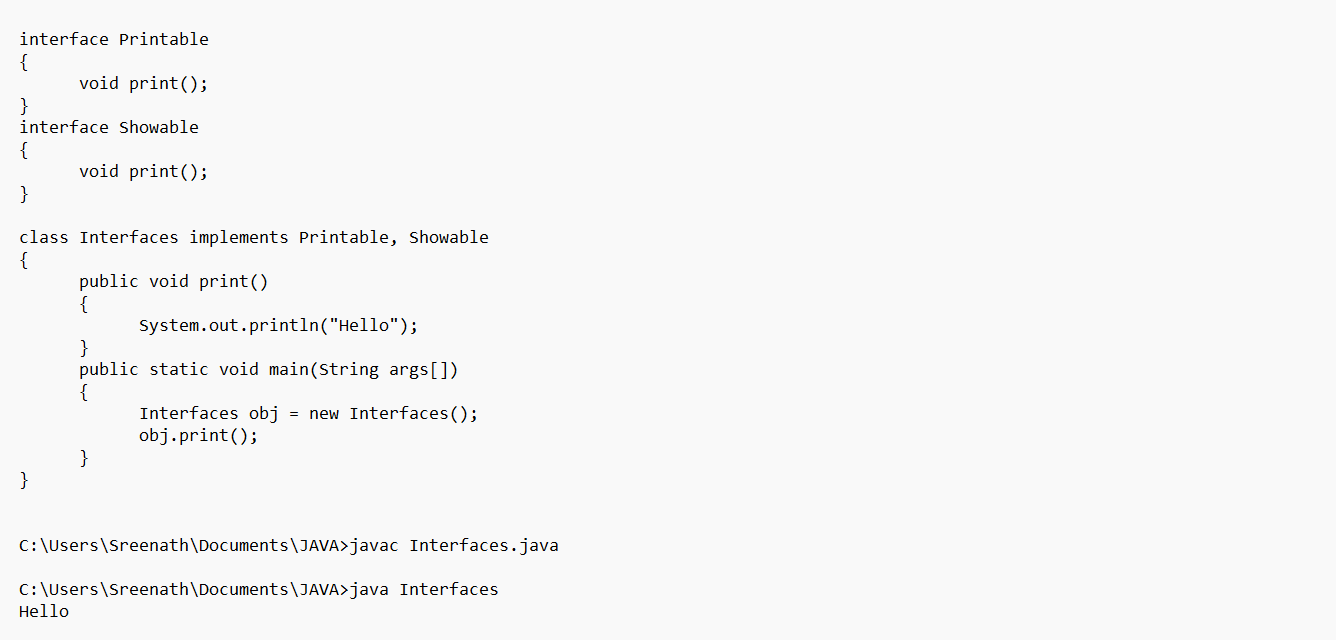
Example:



Multiple inheritance is not supported through java class, but it is possible by an interface, why?

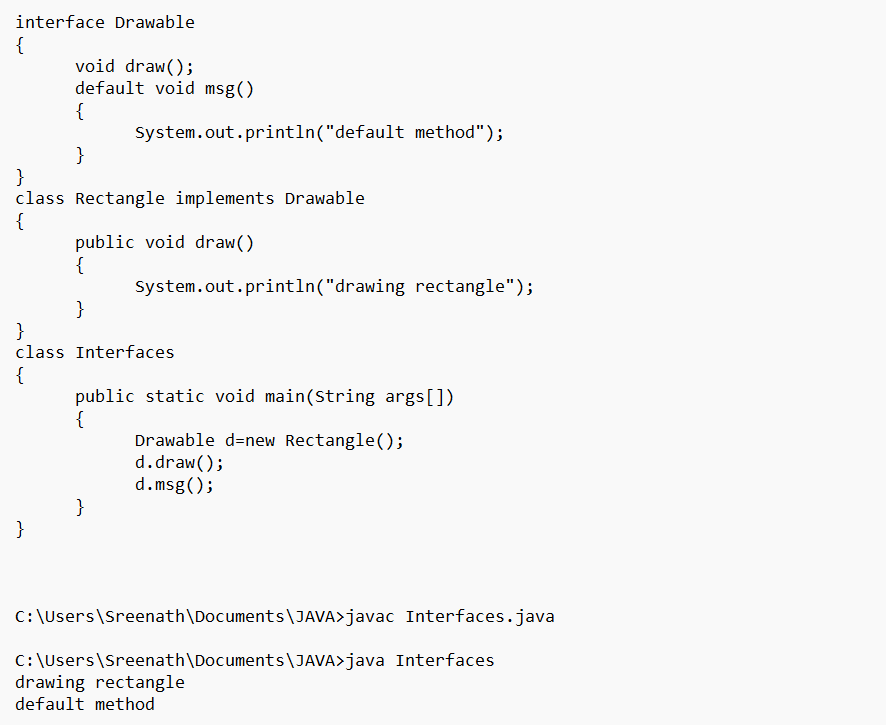
Multiple inheritance is not supported in the case of class because of ambiguity.

However, it is supported in case of an interface because there is no ambiguity. It is because its implementation is provided by the implementation class.

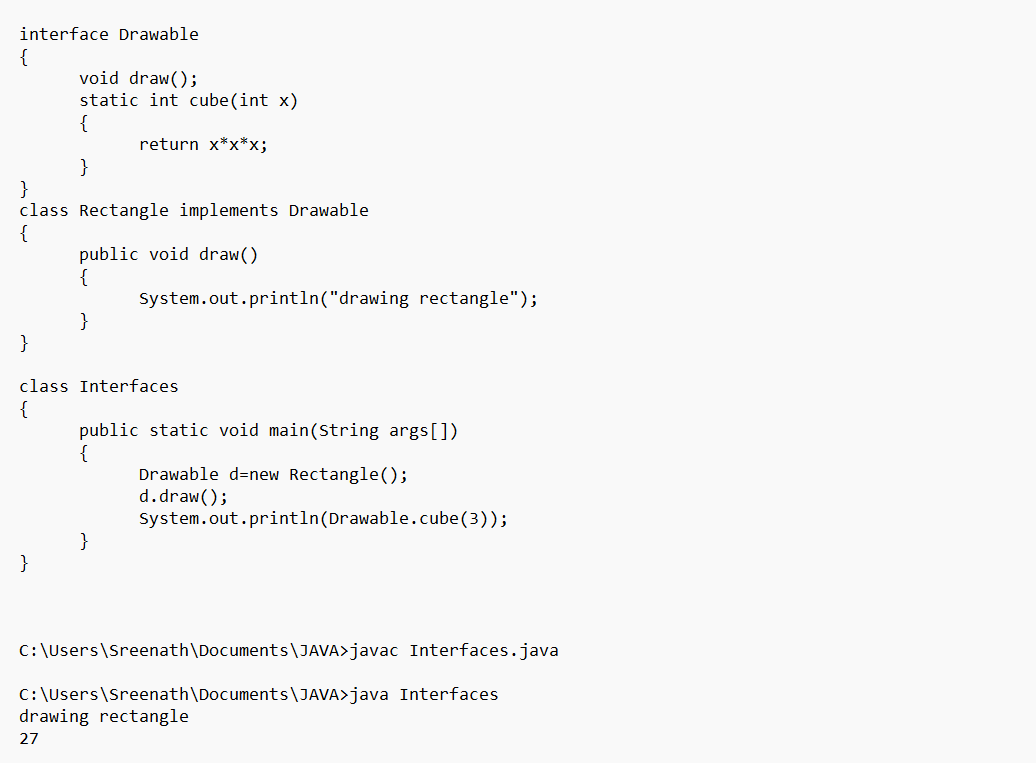




Default Method in Interface



Static Method in Interface



Types of Interfaces

1. Normal (can have more than one Abstract Method)
2. Functional Interface / Single Abstract Method
3. Marker Interface (zero Abstract Methods)