Interface

A device or program enabling a user to communicate with a computer.

OR

An interface looks like a class, but has no implementation. The only thing it contains are declarations of events, indexers, methods and/or properties. The reason interfaces only provide declarations is because they are inherited by classes that must provide an implementation for each interface member declared.

OR

Interface in C# is a blueprint of a class. It is like abstract class because all the methods which are declared inside the interface are abstract methods. It cannot have method body and cannot be instantiated. The implementation of the methods is done in the class that implements the interface.

Interfaces specify what a class must do and not how. In C#, an interface can be defined using the **interface** keyword.

Real World Example

Suppose your parents gives you a list of items to purchase, that list is an Interface that you will implement at time of purchasing. Before implementing anything, you list out what you have to done.

Difference between Abstract class and Interface

* Interface definition begins with a keyword interface so it is of type interface
* Abstract classes are declared with the abstract keyword so it is of type class
* Interface has no implementation, but they have to be implemented.
* Abstract class’s methods can have their own default implementations and they may be extended.
* Interface supports multiple inheritance.
* Abstract classes doesn’t support multiple inheritance.

Application of an Interface

Microsoft .Net does not support multiple inheritance. The biggest problem with multiple inheritance is that it allows for ambiguity when the compiler needs to find the correct implementation of a virtual method. Microsoft introduce Interface as a solution for multiple inheritance. Microsoft .NET allows you to have a single class in the list of parents , therefore supporting single inheritance, but you are free to implement as many interfaces as you wish.

Syntax of an Interface

Interface InterfaceName {

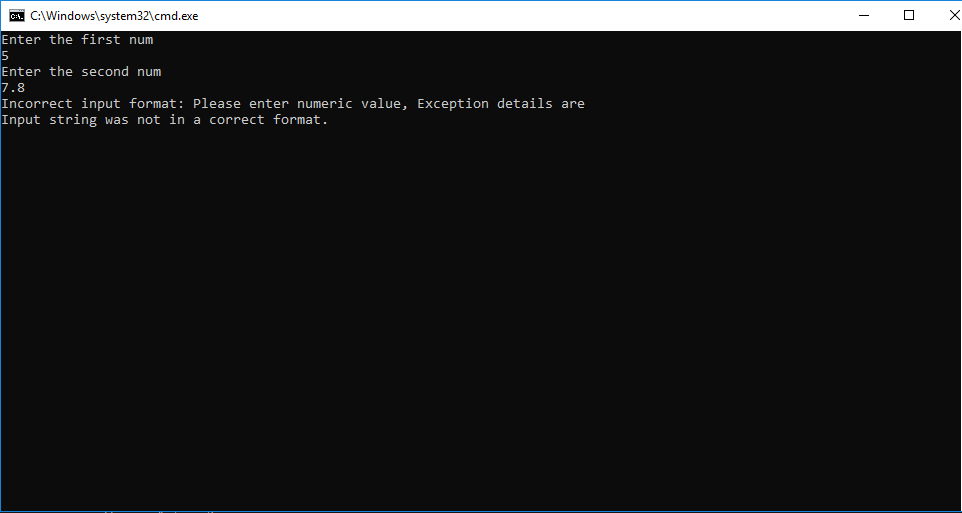
void MethodToImplement(); //Abstract Method signature.

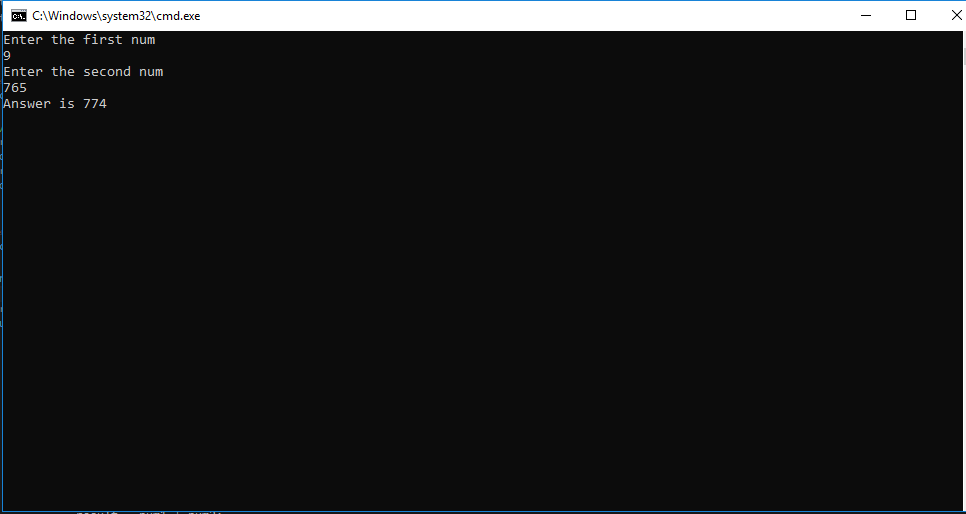
}

Program



Output

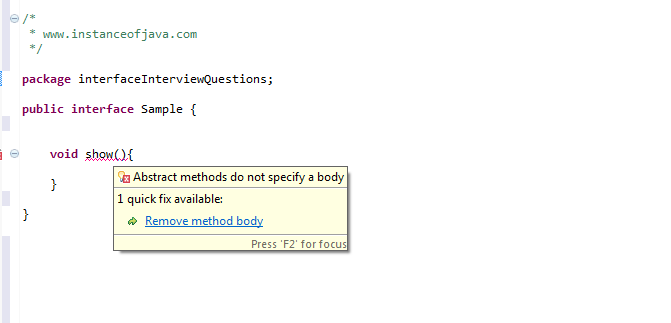




Questions and Answers

**1.What will happen if we define a concrete method in an interface?**

* By default interface methods are abstract.
* if we declare any concrete method in an interface compile time error will come.
* Error:Abstract methods do not specify a body.

[](https://4.bp.blogspot.com/-WlTDjb2dQVg/VtWXfP3-RDI/AAAAAAAAAng/t1gYP8Ll6ak/s1600/interface+concrete+method.png)

**2.Can we create non static variables in an interface?**

* No.We can not create non static variables in an interface.
* If we try to create non static variables compile time error comes.
* By default members will be treated as public static final variables so it expects some value to be initialized.

1. package com.instanceofjava;
2. interface JavaInterface{
4. int x, y; // compile time error
5. }

**3.What will happen if we not initialize variables in an interface.**

* Compile time error will come because by default members will be treated as public static final variables so it expects some value to be initialized.

1. package com.instanceofjava;
2. interface JavaInterface{
4. int x, y; // compile time error: The blank final field y may not have been initialized
5. }

**4.Can we declare interface members as private or protected?**

* No.

1. package com.instanceofjava;
2. interface JavaInterface{
4. private int x; // compile time error: Illegal modifier for the interface field Sample.x; only
5. public, static & final are permitted
6. protected int a; // compile time error: Illegal modifier for the interface field Sample.a; only
7. public, static & final are permitted
8. }

**5.When we need to use extends and implements?**

* A class will implements an interface.
* A class will extends another class.
* An interface extends another interface.

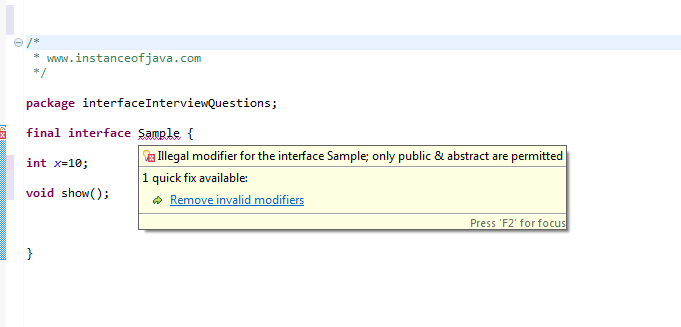
**6.Can we create object for an interface?**

* NO. We can not create object for interface.
* We can create a variable for an interface.

1. package com.instanceofjava;
2. interface JavaInterface{
4. void show();
5. }
6. package com.instanceofjava;
7. interface A implements JavaInterface {
9. void show(){
10. // code
11. }
12. public static void main(String args[]){
14. JavaInterface obj= new JavaInterface(); // Error: Cannot instantiate the type JavaInterface
16. }
17. }

**7.Can we declare interface as final?**

* No. Compile time error will come.
* Error: Illegal modifier for the interface Sample; only public & abstract are permitted.

[](https://3.bp.blogspot.com/-Os3PkMKXd4g/VtWhsViGPQI/AAAAAAAAAnw/vSUj5KzjNO4/s1600/interface+final.png)

**8.Can we declare constructor  inside an interface?**

* No. Interfaces does not allow constructors.
* The variables inside interfaces are static final variables means constants and we can not create object for interface so there is no need of constructor in interface that is the reason interface doesn't allow us to create constructor.

[](https://3.bp.blogspot.com/-7C_DF5VxRAs/Vs7bWEG5dVI/AAAAAAAAAm8/vnVVgGOZu4Y/s1600/interface+constructor.png)

**9.What will happen if we are not implementing all the methods of an interface in class which implements an interface?**

* A class which implements an interface should implement all the methods (abstract) otherwise compiler will throw an error.
* The type Example must implement the inherited abstract method JavaInterface.show()
* If we declare class as abstract no need to implement methods.
* No need of overriding default and static methods.

1. package com.instanceofjava;
2. interface JavaInterface{
4. void show();
5. package com.instanceofjava;
6. interface A implements JavaInterface { // The type Example must implement the inherited
7. abstract method JavaInterface.show()
9. public static void main(String args[]){
11. }
12. }

Reference

* <http://www.instanceofjava.com/2016/03/java-interface-interview-questions.html>
* <https://www.tutorialspoint.com/csharp/csharp_interfaces.htm>
* <http://net-informations.com/faq/net/interface.htm>
* <https://www.codeguru.com/csharp/csharp/cs_syntax/interfaces/article.php/c7563/Interfaces-in-C.htm>