303626-2-8E AID:258164 | 22/08/2020

**Depreciation:**

**Java interface :** Interface are the blueprint of the class.it specify what class must do and not how. It is use to achieve abstraction. It support multiple inheritance. It can be used to achieve loose coupling. Loose coupling is a design goal that seeks to reduce the inter dependencies between components of system with goal of reducing the risk that change in one components will require change in any other components. A class implements an interface, thereby inheriting the abstract methods of interface. Method bodies exist only for default method and static methods . Unless the class that implements the interface is abstract , all the methods of the interface need to be defined in the class. Class and interface are both similar.

what happen if java interface specifies a particular method signature and a class that implements the interface provide different signature for that method ,this is not possible.

This can generate compile error. This called overlapping method signature. Methods with the same name and parameter signature these is risky procedure. This can be solve by override method belong to interface. These Java specification does not give any solution to this problem.

**The explanation of code which is given in example using the depreciation:**

In this example , the java interface SampleInterface and java class SampleClass has been same naming signature SampleInterface . This can be compiler error.

**The following shows the some mirror changes in code and get error free code:**

Public interface SampleInterface

Public int sampleMethod();

}

And the class SampleClass is

Public class SampleClass implements SampleInterface , SampleInterface\_2 /\* create another interface too\*/

{

Public Boolean sampleMethod()

{

Return true;

}

}

{

Create another insterface too.Use multiple interface for these example . The java interface and java class has been two different naming signature.