**Default Method and Multiple Inheritance**

The [**multiple inheritance**](https://beginnersbook.com/2013/05/java-multiple-inheritance/) problem can occur, when we have two interfaces with the default methods of same signature. Lets take an example.

interface MyInterface{

 default void newMethod(){

 System.out.println("Newly added default method");

 }

 void existingMethod(String str);

}

interface MyInterface2{

 default void newMethod(){

 System.out.println("Newly added default method");

 }

 void disp(String str);

}

public class Example implements MyInterface, MyInterface2{

 // implementing abstract methods

 public void existingMethod(String str){

 System.out.println("String is: "+str);

 }

 public void disp(String str){

 System.out.println("String is: "+str);

 }

 public static void main(String[] args) {

 Example obj = new Example();

 //calling the default method of interface

 obj.newMethod();

 }

}

Output:

Error: Duplicate default methods named newMethod with the parameters () and () are inherited from the types MyInterface2 and MyInterface

This is because we have the same method in both the interface and the compiler is not sure which method to be invoked.

**How to solve this issue?**
To solve this problem, we can implement this method in the implementation class like this:

interface MyInterface{

 default void newMethod(){

 System.out.println("Newly added default method");

 }

 void existingMethod(String str);

}

interface MyInterface2{

 default void newMethod(){

 System.out.println("Newly added default method");

 }

 void disp(String str);

}

public class Example implements MyInterface, MyInterface2{

 // implementing abstract methods

 public void existingMethod(String str){

 System.out.println("String is: "+str);

 }

 public void disp(String str){

 System.out.println("String is: "+str);

 }

 //Implementation of duplicate default method

 public void newMethod(){

 System.out.println("Implementation of default method");

 }

 public static void main(String[] args) {

 Example obj = new Example();

 //calling the default method of interface

 obj.newMethod();

 }

}