1) define interface "First" with "default void fun()" and interface "Second" with "default void fun()".

Now derive a class Child from both these interfaces.

Inside main function instantiate Child class instance and invoke "fun" method.

interface First{

default void fun() {

System.***out***.println("First fun");

}

}

interface Second{

default void fun() {

System.***out***.println("Second fun");

}

}

public class Child implements First,Second{

public void fun() {

System.***out***.println("Child class fun");

}

public static void main(String[] args) {

Child c= new Child();

c.fun();

}

}

2) define interface "Third" with "default void disp1()" and "static void disp2()" methods.

derive a class "Sub" from "Third" ( do not override "disp1")

inside main function invoke "disp1" and "disp2" methods.

interface Third{

default void disp1() {

System.***out***.println("Third default disp1");

}

static void disp2() {

System.***out***.println("Third static disp2");

}

}

public class Sub implements Third{

public static void main(String[] args) {

Sub s=new Sub();

s.disp1();

Third.*disp2*();

}

}

3)define interface "Fourth" with "default void disp3()" method.

define a class "Base1" with "public void disp3()" method,

now derive a class Sub1 from "Base1" and "Fourth". Sub1 class should have only "public void myfun()" method.

inside main create an object of "Sub1" and invoke "disp3".

once you get the result, also try to invoke "disp3" of "Fourth"

interface Fourth{

default void disp3() {

System.***out***.println("Fourth default disp3");

}

}

class Base1 {

public void disp3() {

System.***out***.println("Base1 default disp3");

}

}

public class Sub1 extends Base1 implements Fourth{

public void myfun() {

System.***out***.println("Sub 1 myfun");

}

public static void main(String[] args) {

Sub1 s = new Sub1();

s.disp3();

}

}

4) define interface "Base1" with "default void fun1()".

derive an interface "Sub1" from "Base1" with "default void fun2()"

derive a class "Sub2" from "Sub1". It should have only "public void fun3()" method.

inside main function, create an object of "Sub2" and make sure when u invoke "fun3", it should call "Sub1's fun2" which should call "Base1's fun1" method.

interface Base1{

default void fun1() {

System.***out***.println("default void fun1");

}

}

interface Sub1 extends Base1{

default void fun2() {

System.***out***.println("default void fun2");

Base1.super.fun1();

}

}

public class Sub2 implements Sub1{

public void fun3() {

System.***out***.println("public void fun3");

fun2();

}

public static void main(String[] args) {

Sub2 s = new Sub2();

s.fun3();

}

}

5) define interface "First" with "static void disp1()" method.

class "Base" with "static void disp2()" method.

derive a class "Sub" from "Base" and "First"

now define a class Demo in which define "main" method.

inside main show how many ways you can invoke "disp1" and "disp2" methods.

interface First{

static void disp1() {

System.***out***.println("static void disp1()");

}

}

class Base{

static void disp2() {

System.***out***.println("static void disp2()");

}

}

class Sub extends Base implements First{}

public class Demo {

public static void main(String[] args) {

Sub s = new Sub();

s.*disp2*();

Sub.*disp2*();

First.*disp1*();

}

}