**Child 1 class**

package com.abstractionAndinterface.demo;

public class Child1 extends Child2 implements Interface1,Interface2 {

public void m1()

{

System.out.println(x);

}

public void m2()

{

System.out.println(y);

}

public static void main(String[] args) {

Child1 c1 = new Child1();

c1.m1();

c1.m2();

c1.m3();

}

}

**Child 2 class**

package com.abstractionAndinterface.demo;

public class Child2 {

int z=300;

void m3()

{

System.out.println(z);

}

}

**Interface 1 class**

package com.abstractionAndinterface.demo;

public interface Interface1 {

int x=100;

void m1();

}

**Interface 2 class**

package com.abstractionAndinterface.demo;

public interface Interface2 {

int y=200;

void m2();

}

**InterfaceDemo class**

package com.abstractionAndinterface.demo;

import java.awt.Rectangle;

/\* Interface :

\* An interface is a blue print of class.

\* Interface contains final & static keyword

\* Interface contains abstract methods.( also allow default and static method java8 onwards)

\* An abstract method is a method contains signature but not body

\* methods in interface are public

\* Interface supports the functionality of multiple inheritance

\* we can define interface with interface keyword

\* A class extends another class, an interface extends another interface but class implements an interface

\* we can create object reference for interface but we cannot instantiate interface.

\*

\*/

interface Shape

{

int length=10; // final & static variable

int width =20; // final & static variable

void circle(); // abstract method

default void square() // default method

{

System.out.println("This is default method..");

}

static void rectangle()

{

System.out.println("This is static method..");

}

}

public class InterfaceDemo implements Shape{

public void circle()

{

System.out.println("This is abstract method..");

}

public static void main(String[] args) {

//scenario1

//InterfaceDemo idobj = new InterfaceDemo();

//idobj.circle(); // abstract method

//idobj.square(); // default method

//Shape.rectangle(); // static method can directly access from interface

//scenario2

Shape sh = new InterfaceDemo();

sh.circle();

sh.square();

Shape.rectangle();

}

}