Write code (preferably test automation example) to implement inheritance using interfaces, abstract classes

Java supports multiple interface inheritance where an interface extends more than one super interfaces.

The following is the syntax used to extend multiple interfaces in Java:

*access\_specifier interface subinterfaceName extends superinterface1, superinterface2, …… {*

*// Body  
}*

The following is an example which implements the multiple inheritances in interfaces:

|  |
| --- |
| // Java program to demonstrate  // the multiple inheritance  // in interface    // Interface to implement the  // addition and subtraction methods  interface Add\_Sub {      public void add(double x, double y);      public void subtract(double x, double y);  } |

// Interface to implement the

// multiplication and division

interface Mul\_Div {

    public void multiply(double x, double y);

    public void divide(double x, double y);

}

// Calculator interface which extends

// both the above defined interfaces

interface Calculator extends Add\_Sub, Mul\_Div {

    public void printResult(double result);

}

// Calculator class which

// implements the above

// interface

public class MyCalculator implements Calculator {

    // Implementing the addition

    // method

    public void add(double x, double y)

    {

        double result = x + y;

        printResult(result);

    }

    // Implementing the subtraction

    // method

    public void subtract(double x, double y)

    {

        double result = x - y;

        printResult(result);

    }

    // Implementing the multiplication

    // method

    public void multiply(double x, double y)

    {

        double result = x \* y;

        printResult(result);

    }

    // Implementing the division

    // method

    public void divide(double x, double y)

    {

        double result = x / y;

        printResult(result);

    }

    // Implementing a method

    // to print the result

    public void printResult(double result)

    {

        System.out.println(

            "The result is : " + result);

    }

    // Driver code

    public static void main(String args[])

    {

        // Creating the object of

        // the calculator

        MyCalculator c = new MyCalculator();

        c.add(5, 10);

        c.subtract(35, 15);

        c.multiply(6, 9);

        c.divide(45, 6);

    }

}

Inheritance using abstract classes

In Java, we can have an abstract class without any abstract method. This allows us to create classes that cannot be instantiated but can only be inherited.

// An abstract class without any abstract method

abstract class Base {

    void fun() { System.out.println("Base fun() called"); }

}

class Derived extends Base {

}

class Main {

    public static void main(String args[])

    {

        Derived d = new Derived();

        d.fun();

    }

}