

## **Program 8: Develop programs for implementation of**

**a) Single inheritance**

**b) Multiple inheritance**

### **A) SINGLE INHERITANCE:**

#### **CODE:**

```
class Base{

    int x = 3;

    void meth1(){

        System.out.println("base class var: "+x);

    }

}

class Derived extends Base {

    int y =4;

    void meth2(){

        System.out.println("derived class var: "+y);

    }

}

class Single_inheritance{

    public static void main(String[] args) {
```

NAME : MINAL CHHATRE  
ENROLL NO: 1906016  
G3 (A)

```
    Derived d1 = new Derived();

    d1.meth1();

    d1.meth2();

    Base b1 = new Base();

    b1.meth1();

    //    b1.meth2(); not possible

    }

}
```

### **OUTPUT:**

D:\java>javac Single\_inheritance.java

D:\java>java Single\_inheritance

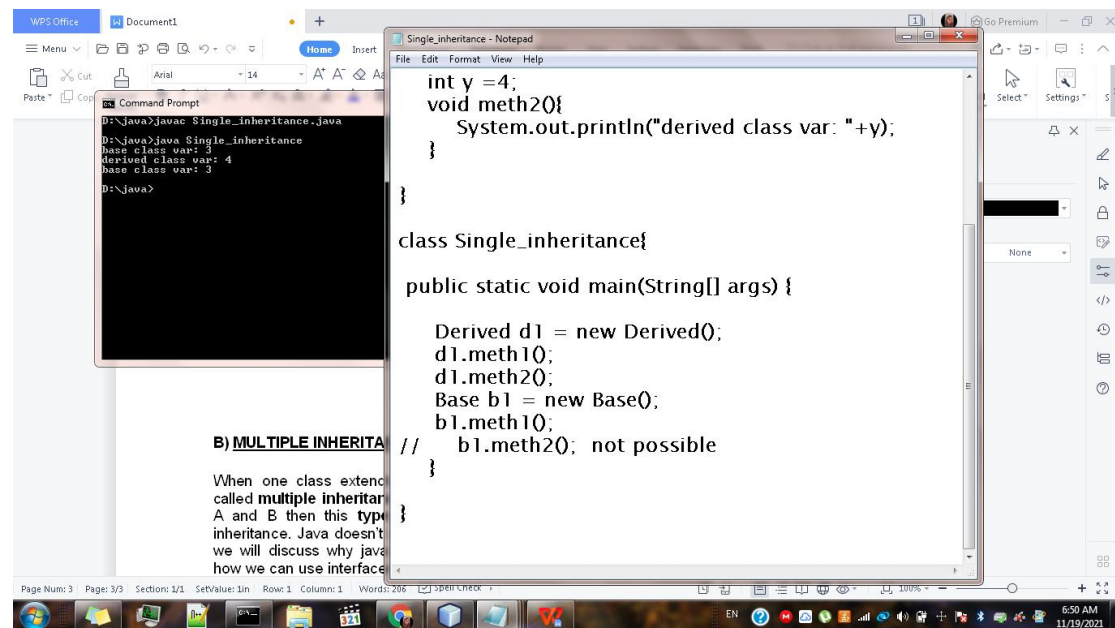
base class var: 3

derived class var: 4

base class var: 3

D:\java>

NAME : MINAL CHHATRE  
ENROLL NO: 1906016  
G3 (A)



The screenshot shows a Windows desktop with a Notepad window titled 'Single\_inheritance - Notepad' and a Command Prompt window. The Notepad window contains the following Java code:

```
int y = 4;
void meth20{
    System.out.println("derived class var: "+y);
}

class Single_inheritance{
    public static void main(String[] args) {
        Derived d1 = new Derived();
        d1.meth10();
        d1.meth20();
        Base b1 = new Base();
        b1.meth10();
        // b1.meth20(); not possible
    }
}
```

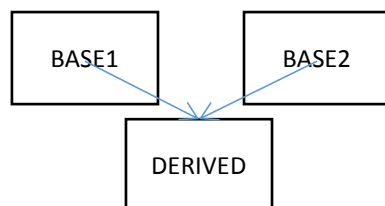
The Command Prompt window shows the output of the code:

```
D:\java>javac Single_inheritance.java
D:\java>java Single_inheritance
Base class var: 3
Derived class var: 4
Base class var: 3
D:\java>
```

Below the Command Prompt window, there is a section titled 'B) MULTIPLE INHERITANCE' with a brief explanation: 'When one class extends more than one classes then this is called multiple inheritance. Java doesn't allow multiple inheritance. In this article, we will discuss why java doesn't allow multiple inheritance and how we can use interfaces instead of classes to achieve the same purpose.'

## **B) MULTIPLE INHERITANCE:**

When one class extends more than one classes then this is called **multiple inheritance**. For example: Class C extends class A and B then this **type of inheritance** is known as multiple inheritance. Java doesn't allow multiple inheritance. In this article, we will discuss why java doesn't allow multiple inheritance and how we can use interfaces instead of classes to achieve the same purpose.



## **Why Java doesn't support multiple inheritance?**

C++ , Common lisp and few other languages supports multiple inheritance while java doesn't support it. Java doesn't allow multiple inheritance to **avoid the ambiguity** caused by it. One of the example of such problem is the **diamond problem** that occurs in multiple inheritance.

NAME : MINAL CHHATRE  
ENROLL NO: 1906016  
G3 (A)

**CODE:**

```
// Java Program to Illustrate Unsupportance of
// Multiple Inheritance

// Importing input output classes
import java.io.*;

// Class 1
// First Parent class
class Parent1 {

    // Method inside first parent class
    void fun() {

        // Print statement if this method is called
        System.out.println("Parent1");
    }
}

// Class 2
// Second Parent Class
class Parent2 {

    // Method inside first parent class
    void fun() {

        // Print statement if this method is called
        System.out.println("Parent2");
    }
}

// Class 3
// Trying to be child of both the classes
class Test extends Parent1, Parent2 {

    // Main driver method
    public static void main(String args[]) {

        // Creating object of class in main() method
        Test t = new Test();

        // Trying to call above functions of class where
        // Error is thrown as this class is inheriting
        // multiple classes
        t.fun();
    }
}
```

```
GFG.java:228: error: '{' expected
class Test extends Parent1, Parent2 {
                        ^
1 error
```

NAME : MINAL CHHATRE  
ENROLL NO: 1906016  
G3 (A)

CODE:

//MULTIPLE INHERITANCE USING INTERFACE

```
class Student{
    int m1,m2;
    void getmarks(int x,int y){
        m1=x;
        m2=y;
    }
    void putmarks() {
        System.out.println("M1: "+m1);
        System.out.println("M2: "+m2);
    }
}
interface Sport{
    int sp=6;
    void spmarks();
}
class Result extends Student implements Sport{
    public void spmarks() {
        System.out.println("Sport marks "+sp);
    }
    void display() {
        putmarks();
        spmarks();
        int total = m1+m2+sp;
        System.out.println("Total : "+total);
    }
}
class Interface_pg {
    public static void main(String[] args) {
        Result obj = new Result();
        obj.getmarks(80,60);
        obj.display();
    }
}
```

NAME : MINAL CHHATRE  
ENROLL NO: 1906016  
G3 (A)

## OUTPUT:

```
D:\>cd java
```

```
D:\java>javac Interface_pg.java
```

```
D:\java>java Interface_pg
```

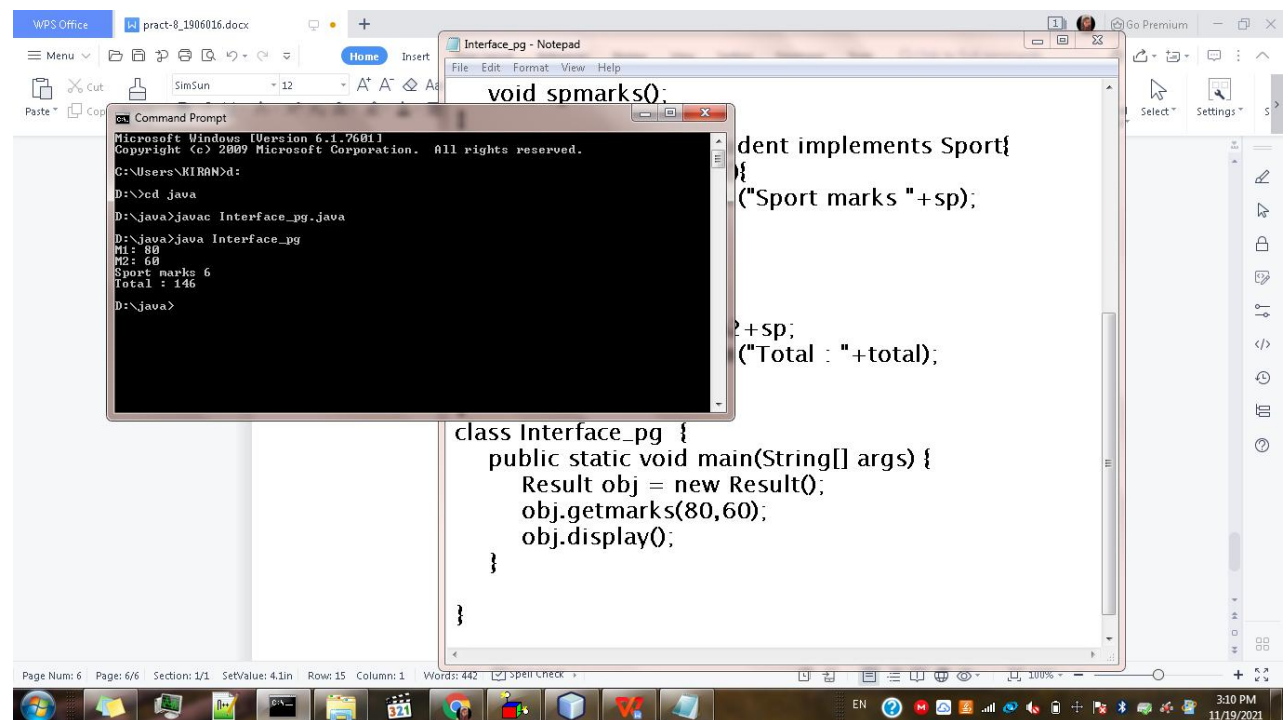
M1: 80

M2: 60

Sport marks 6

Total : 146

```
D:\java>
```



The screenshot displays a Windows desktop environment. In the background, a WPS Office window is open with a document titled 'pract-8\_1906016.docx'. Overlaid on this is a Command Prompt window showing the following commands and output:

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\KIRAN>cd
D:\>cd java
D:\java>javac Interface_pg.java
D:\java>java Interface_pg
M1: 80
M2: 60
Sport marks 6
Total : 146
D:\java>
```

In the foreground, a Notepad window titled 'Interface\_pg - Notepad' displays the source code for 'Interface\_pg.java':

```
void spmarks0:
    dent implements Sport{
    }
    ("Sport marks "+sp);

    2+sp;
    ("Total : "+total);

class Interface_pg {
    public static void main(String[] args) {
        Result obj = new Result();
        obj.getmarks(80,60);
        obj.display();
    }
}
```