# Program 8: Develop programs for implementation of

# Single inheritance

# b) Multiple inheritance

1. **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) {

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

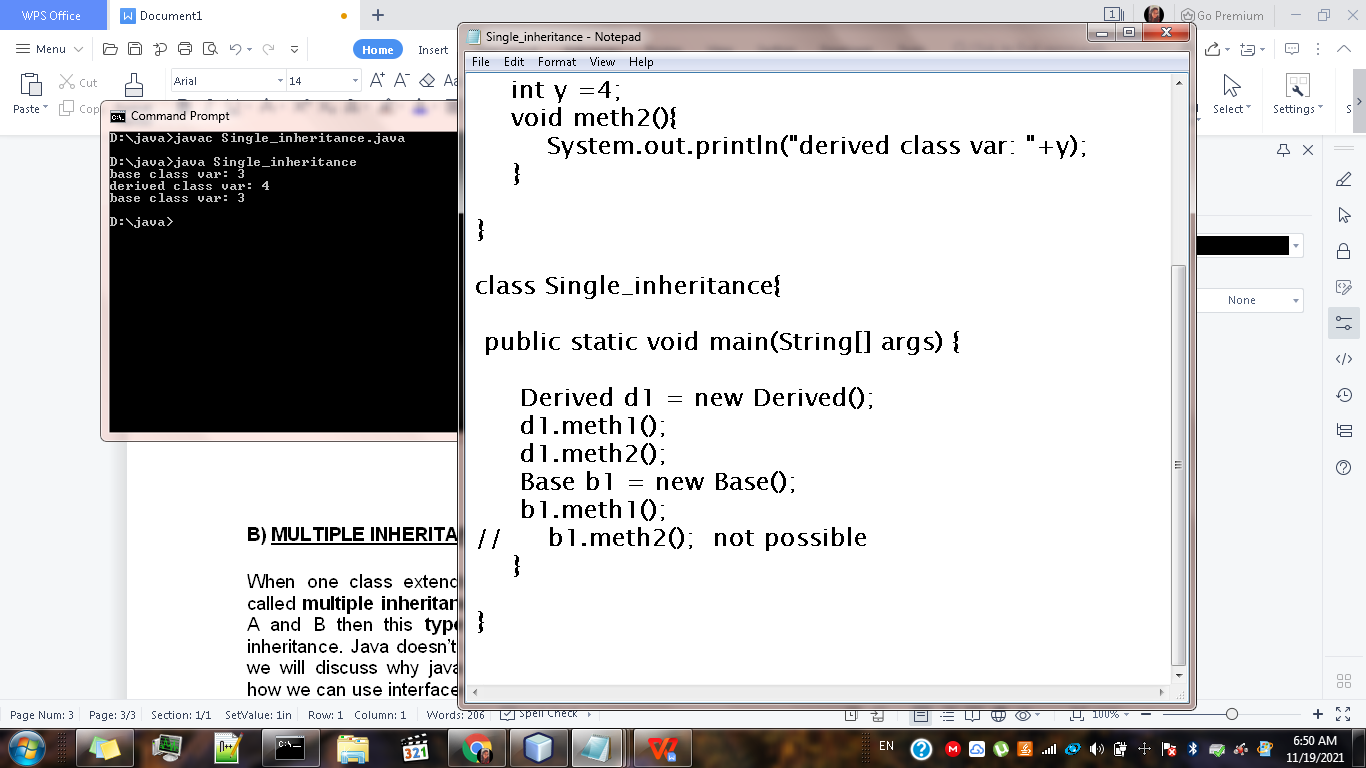
D:\java>java Single\_inheritance

base class var: 3

derived class var: 4

base class var: 3

D:\java>



1. **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](https://beginnersbook.com/2013/05/java-inheritance-types/)** 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.

BASE2

BASE1

DERIVED

## **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.

**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

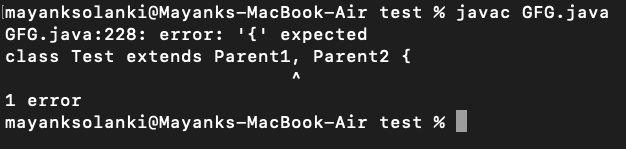
    // Error is thrown as this class is inheriting

    // multiple classes

    t.fun();

  }

}



**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();

}

}

**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>

