class Calculation{

int z;

public void addition(int x, int y){

z = x+y;

System.out.println("The sum of the given numbers:"+z);

}

public void Substraction(int x,int y){

z = x-y;

System.out.println("The difference between the given numbers:"+z);

}

}

public class My\_Calculation extends Calculation{

public void multiplication(int x, int y){

z = x\*y;

System.out.println("The product of the given numbers:"+z);

}

public static void main(String args[]){

int a = 20, b = 10;

My\_Calculation demo = new My\_Calculation();

demo.addition(a, b);

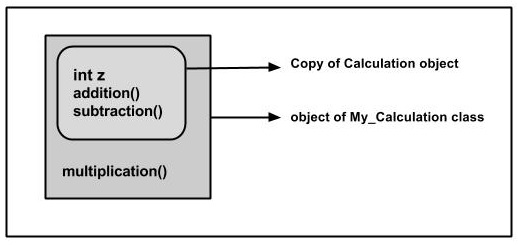
demo.Substraction(a, b);

demo.multiplication(a, b);

}

}

In the given program when an object to **My\_Calculation** class is created, a copy of the contents of the super class is made with in it. That is why, using the object of the subclass we can access the members of a super class.



The Superclass reference variable can hold the subclass object, but using that variable we can access only the members of the superclass, so to access the members of both classes it is recommended to always create reference variable to the subclass.

If you consider the above program we can instantiate the class as given below as well. But using the superclass reference variable ( **cal** in this case ) we cannot call the method **multiplication()**, which belongs to the subclass My\_Calculation.

Calculation cal = new My\_Calculation();

demo.addition(a, b);

demo.Subtraction(a, b);

**Note** − A subclass inherits all the members (fields, methods, and nested classes) from its superclass. Constructors are not members, so they are not inherited by subclasses, but the constructor of the superclass can be invoked from the subclass.