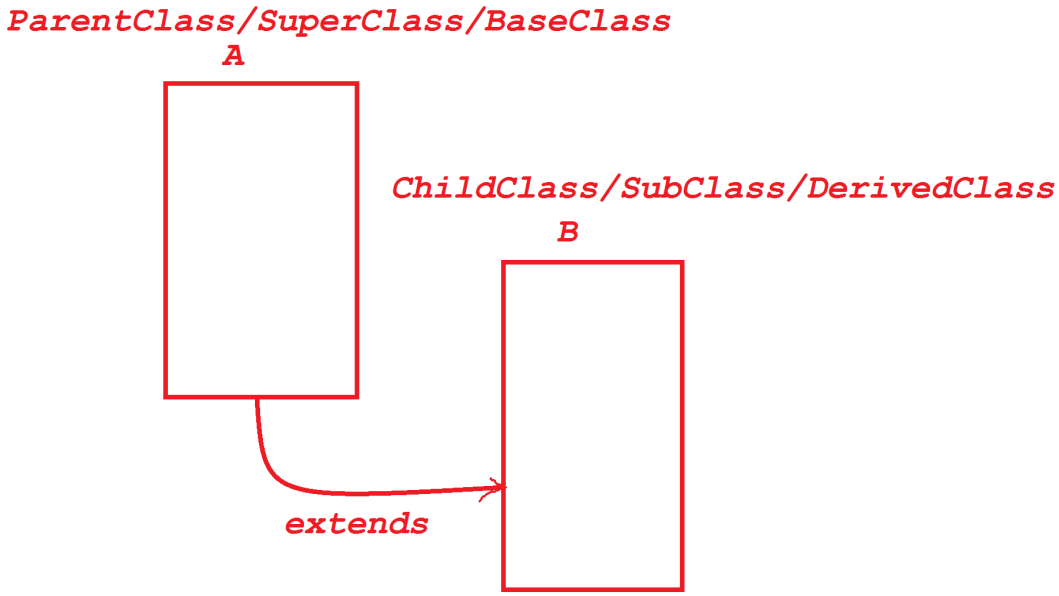
Dt : 17/8/2023

**imp*

2.Inheritance process in Java:

=>The process of interlinking classes using "extends" keyword is known as inheritance process.

Diagram:



syntax:

```
class A
{
 //Class_body
```

}

class B extends A

{

//Class_body

}

=>In this process the members of Class-A are available to Class-B,and Class-A is known as ParentClass and Class-B is known as ChildClass.

=>In inheritance process we always create object for ChildClass,because the ChildClass object will hold the members of ParentClass and ChildClass.

==========

Case-1 : Variables and methods from the ParentClass/SuperClass

Project_name : Inheritance_App1

packages,

p1 : A.java

```
package p1;
public class A {
    public int a;
    public static int b;
    public void m1() {
         System.out.println("===ParentClass-m1()====");
         System.out.println("The value a:"+a);
         System.out.println("The value b:"+b);
    }
```

Venkatesh Maipathii

```
    public static void m2() {
         System.out.println("===ParentClass-m2()====");
         //System.out.println("The value a:"+a);
         System.out.println("The value b:"+b);
    }
}
```

p1 : B.java

```
package p1;
public class B extends A{
    public int x;
    public static int y;
    public void m3() {
        System.out.println("===ChildClass-m3()====");
        System.out.println("The value x:"+x);
        System.out.println("The value y:"+y);
    }
    public static void m4() {
         System.out.println("===ChildClass-m4()====");
         //System.out.println("The value x:"+x);
         System.out.println("The value y:"+y);
    }
}
```

p2 : DemoInheritance1.java(MainClass)

```
package p2;

import p1.*;

import java.util.Scanner;

public class DemoInheritance1 {

        public static void main(String[] args) {

    Scanner s = new Scanner(System.in);

    B ob = new B();//ChildClass-Object
```

Venkatesh Maipathii

```
    System.out.println("Enter the value a:");
    ob.a = s.nextInt();
    System.out.println("Enter the value b:");
    B.b = s.nextInt();
    System.out.println("Enter the value x:");
    ob.x = s.nextInt();
    System.out.println("Enter the value y:");
    B.y = s.nextInt();

    ob.m1();
    B.m2();
    ob.m3();
    B.m4();
    s.close();
        }
}
```

o/p:

```
Enter the value a:
11
Enter the value b:
12
Enter the value x:
```

Venkatesh Maipathii

13

Enter the value y:

14

===ParentClass-m1()====

The value a:11

The value b:12

===ParentClass-m2()====

The value b:12

===ChildClass-m3()====

The value x:13

The value y:14

===ChildClass-m4()====

The value y:14

Execution flow of above application:

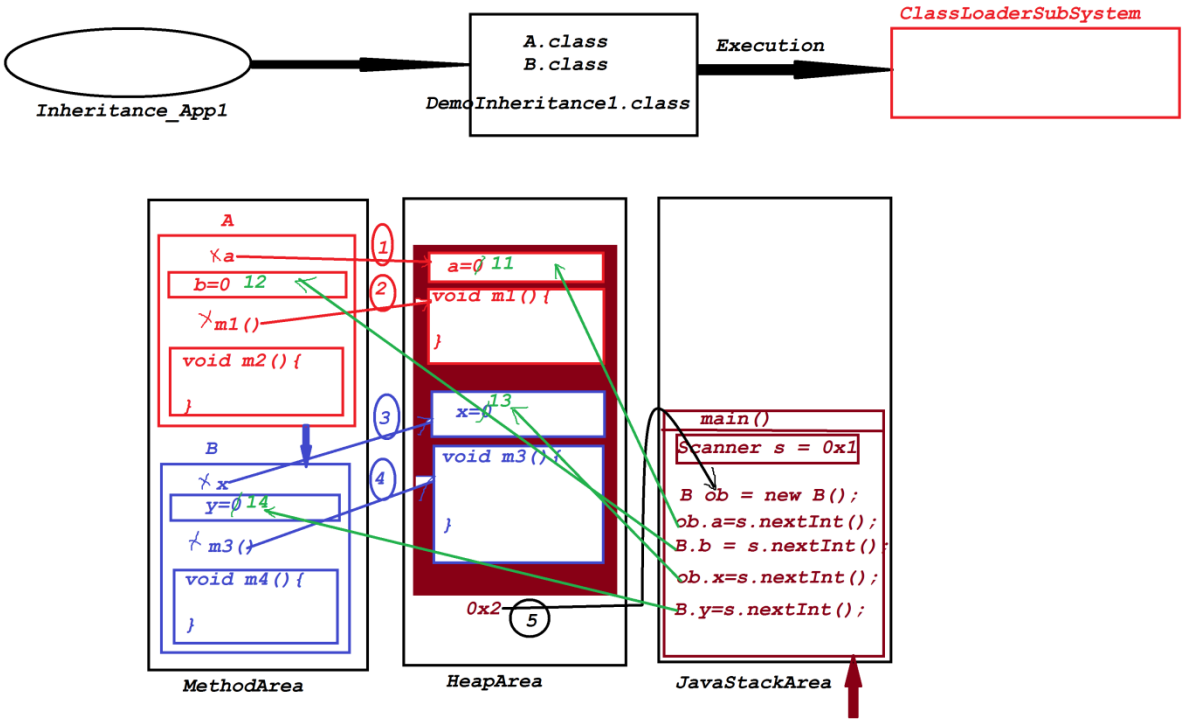
ClassFiles:

A.class

B.class

DemoInheritance1.class(MainClass)

Venkatesh Maipathii



(i)In inheritance process all the variables and methods of PClass are available to ChildClass.

(ii)Instance members of PClass can be accessed with ChildClass object name.

(iii)Static members of PClass can be accessed with ChildClass name.

(iv)In Inheritance Process ParentClass is loaded to MethodArea first and then ChildClass is loaded.

(v)In Inheritance process while Object creation,PClass Instancec members will get the memory first and then CClass Instance members will get the memory.

Venkatesh M

==========

faq:

define Method Overriding process?

=>The method with same method signature in PClass and CClass,then PClass method is replaced by CClass method while object creation process is known as Method Overriding process or Method Replacement process.

=>Same method signature means,

same return_type

same method_name

same para_list

same para_type

Project_name : Inheritance_App2(Demonstrating Instance method Overriding)

packages,

p1 : PClass.java

```
package p1;
public class PClass {
    public void m(int x) {
         System.out.println("====PClass-m()-body====");
         System.out.println("The value x:"+x);
    }
}
```

Venkatesh Maipathi

p1 : CClass.java

```
package p1;
public class CClass extends PClass{
  public void m(int x) {
       System.out.println("====CClass-m()-body====");
       System.out.println("The value x:"+x);
  }
}
```

p2 : DemoInheritance2.java(MainClass)

```
package p2;
import p1.*;
public class DemoInheritance2 {
    public static void main(String[] args) {
        CClass ob = new CClass();//CClass_object
        ob.m(123);
    }
}
```

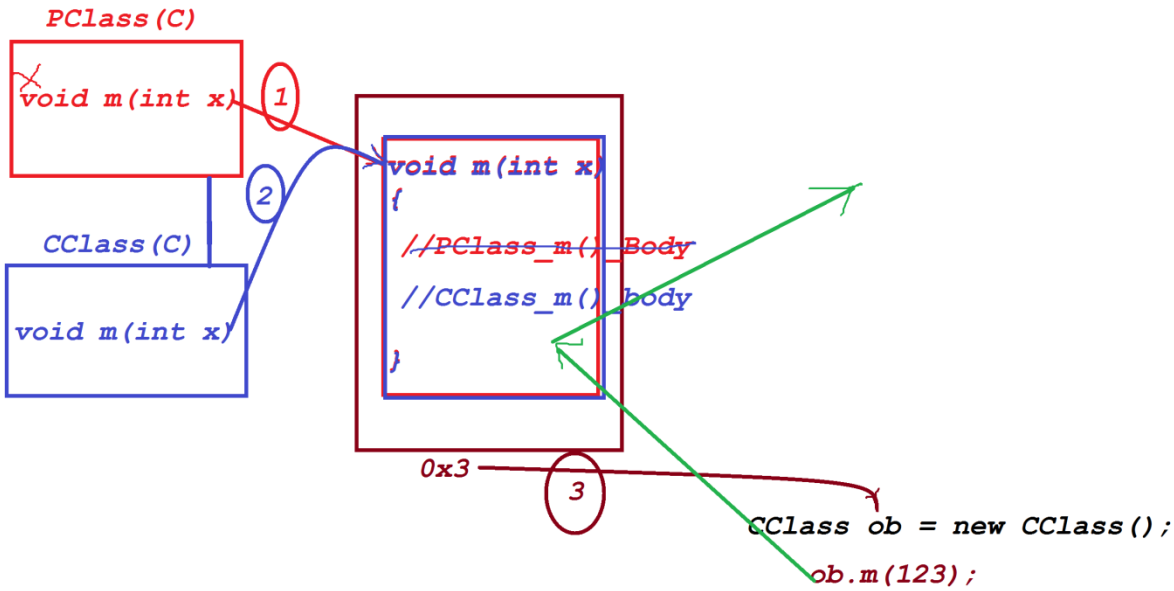
o/p:

```
====CClass-m()-body====

The value x:123
```

Diagram:

Venkatesh Maipathii


PClass(C)
void m(int x)
1
2
CClass(C)
void m(int x)
void m(int x)
{
//PClass_m() Body
//CClass_m() body
}
0x3
3
CClass ob = new CClass();
ob.m(123);

==========

Venkatesh