JAVA PROGRAMMING LAB

**LAB 5**

**Submitted by:** Bharat Kumar

**Enrollment no:** 12814802719

**Topic covered:** Inheritance , Polymorphism , Method overriding

**LAB 5.**1

**Aim:**

* WAP that implements method overriding

**Theory:**

Method Overriding in Java

* [Understanding the problem without method overriding](https://www.javatpoint.com/method-overriding-in-java#moverproblem)
* [Can we override the static method](https://www.javatpoint.com/method-overriding-in-java#movercanstatic)
* [Method overloading vs. method overriding](https://www.javatpoint.com/method-overriding-in-java#moverdiff)

If subclass (child class) has the same method as declared in the parent class, it is known as method overriding in Java.

In other words, If a subclass provides the specific implementation of the method that has been declared by one of its parent class, it is known as method overriding.

### Usage of Java Method Overriding

* Method overriding is used to provide the specific implementation of a method which is already provided by its superclass.
* Method overriding is used for runtime polymorphism

#### Rules for Java Method Overriding

1. The method must have the same name as in the parent class
2. The method must have the same parameter as in the parent class.
3. There must be an IS-A relationship (inheritance).

**Source code:**

class parent{

void show(){

System.out.println("In parent class");

}

}

public class child extends parent{

void show(){

System.out.println("In child class");

}

public static void main(String[] args){

child c=new child();

c.show();

}

}

**Output**



**LAB 4.2**

**Aim:** WAP to illustrate simple inheritance.

**Theory:**

The process by which one class acquires the properties(data members) and functionalities(methods) of another class is called **inheritance**. The aim of inheritance is to provide the reusability of code so that a class has to write only the unique features and rest of the common properties and functionalities can be extended from the another class.  
Child Class:  
The class that extends the features of another class is known as child class, sub class or derived class.

**Parent Class:**  
**The class whose properties and functionalities are used(inherited) by another class is known as parent class, super class or Base class.**

Inheritance is a process of defining a new class based on an existing class by extending its common data members and methods.  
Inheritance allows us to reuse of code, it improves reusability in your java application.  
Note: The biggest advantage of Inheritance is that the code that is already present in base class need not be rewritten in the child class.

This means that the data members(instance variables) and methods of the parent class can be used in the child class as.

If you are finding it difficult to understand what is class and object then refer the guide that I have shared on object oriented programming: [**OOPs Concepts**](https://beginnersbook.com/2013/04/oops-concepts/)

## **Syntax: Inheritance in Java**

To inherit a class we use extends keyword. Here class XYZ is child class and class ABC is parent class. The class XYZ is inheriting the properties and methods of ABC class.

class XYZ extends ABC{}

**Source code:**

class parents{

private String str= "Parent";

private int age=50;

void displayParent(){

System.out.println("Class name: "+str);

System.out.println("Class age: "+age);

}

}  
public class childs extends parents{

private String str;

private int age;

childs(){

str="undefined";

age=-1;

}

childs(String str,int age){

this.str=str;

this.age=age;

}

void displayChild(){

System.out.println("Class name: "+str);

System.out.println("Class age: "+age);

}

public static void main(String[] args){

childs c=new childs("Child",10);

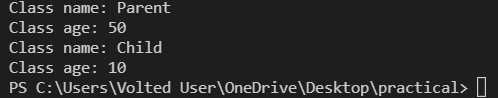
c.displayParent();

c.displayChild();

}

}

**Output:**

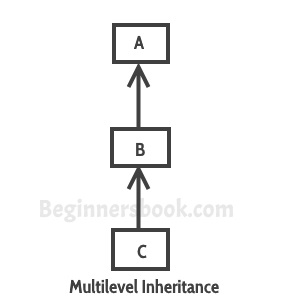


**LAB 4.3**

**Aim:** WAP to illustrate multilevel inheritance.

**Theory:**

# **Multilevel inheritance in java with example**

When a class extends a class, which extends anther class then this is called **multilevel inheritance**. For example class C extends class B and class B extends class A then this [**type of inheritance**](https://beginnersbook.com/2013/05/java-inheritance-types/) is known as multilevel inheritance.  
Lets see this in a diagram:  


It’s pretty clear with the diagram that in Multilevel inheritance there is a concept of grand parent class. If we take the example of this diagram, then class C inherits class B and class B inherits class A which means B is a parent class of C and A is a parent class of B. So in this case class C is implicitly inheriting the properties and methods of class A along with class B that’s what is called multilevel inheritance.

**Source code:**

class grandParent{

private double inheritance=24000.50;

double inheritAmount(){

return inheritance;

}

}

class parent extends grandParent{

private String str= "Parent";

private int age=49;

void displayParent(){

System.out.println("Class name: "+str);

System.out.println("Class age: "+age);

}

double inheritAmount(){

return super.inheritAmount()+5500;

}

}

public class child extends parent{

private String str;

private int age;

child(){

str="undefined";

age=-1;

}

child(String str,int age){

this.str=str;

this.age=age;

}

void displayChild(){

System.out.println("Class name: "+str);

System.out.println("Class age: "+age);

}

double inheritAmount(){

return super.inheritAmount()+20.5;

}

public static void main(String[] args){

child c=new child("Child",10);

c.displayParent();

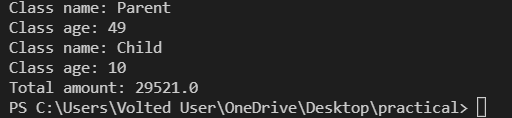
c.displayChild();

System.out.println("Total amount: "+c.inheritAmount());

}

}

**Output:**



**LAB 4.4**

**Aim:** WAP illustrating all uses of super keywords

**Theory:**

# Super Keyword in Java

The super keyword in Java is a reference variable which is used to refer immediate parent class object.

Whenever you create the instance of subclass, an instance of parent class is created implicitly which is referred by super reference variable.

## Usage of Java super Keyword

1. super can be used to refer immediate parent class instance variable.
2. super can be used to invoke immediate parent class method.
3. super() can be used to invoke immediate parent class constructor.



## 1) super is used to refer immediate parent class instance variable.

## 2) super can be used to invoke parent class method

## 3) super is used to invoke parent class constructor.

**Source code:**

class box{

private int a,b,c;

box(){

a=b=c=-1;

}

box(int a,int b,int c){

this.a=a;

this.b=b;

this.c=c;

}

box(box o){

a=o.a;

b=o.b;

c=o.c;

}

float volume(){

return (float)a\*b\*c;

}

void shape(){

System.out.println("It is box shape");

}

}

class boxWeight extends box{

private double w;

// or we can use a for weight too, and further use super.a for superclass a

boxWeight(){

super();

w=-1;

}

boxWeight(int a,int b,int c,double w){

super(a,b,c);

this.w=w;

}

boxWeight(boxWeight wo){

super(wo);

w=wo.w;

}

double density(){

return w/volume();

}

void shape(){

super.shape();

System.out.println("It also have mass factor");

}

}

public class classIntro {

public static void main(String[] args) {

box b1=new box();

box b2=new box(10,15,10);

box b3=new box(b2);

System.out.println(b1.volume()>=0?b1.volume():"No paramteres");

System.out.println(b2.volume()>=0?b2.volume():"No paramteres");

System.out.println(b3.volume()>=0?b3.volume():"No paramteres");

boxWeight b=new boxWeight(5,10,5,25.0);

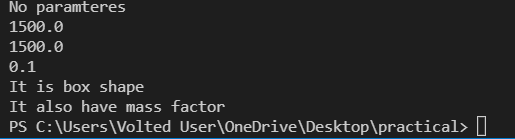
System.out.println(b.density()>=0?b.density():"No paramteres");

b.shape();

}

}

**Output:**



VIVA VOCE QUESTION

1. When will you define a method as static?

Ans. When a method needs to be accessed even before the creation of the object of the class then we should declare the method as static.

2. What are the restriction imposed on a static method or a static block of code?

Ans. A static method should not refer to instance variables without creating an instance and cannot use "this" operator to refer the instance.

3. I want to print "Hello" even before main() is executed. How will you acheive that?

Ans. Print the statement inside a static block of code. Static blocks get executed when the class gets loaded into the memory and even before the creation of an object. Hence it will be executed before the main() method. And it will be executed only once.

4. How-will-you-communicate-between-two-Applets

Ans. The simplest method is to use the static variables of a shared class since there's only one instance of the class and hence only one copy of its static variables. A slightly more reliable method relies on the fact that all the applets on a given page share the same AppletContext. We obtain this applet context as follows: AppletContext ac = getAppletContext(); AppletContext provides applets with methods such as getApplet(name), getApplets(),getAudioClip, getImage, showDocument and showStatus().

5. Which classes can an applet extend?

Ans. An applet can extend the java.applet.Applet class or the java.swing.JApplet class. The java.applet.Applet class extends the java.awt.Panel class and enables you to use the GUI tools in the AWT package. The java.swing.JApplet class is a subclass of java.applet.Applet that also enables you to use the Swing GUI tools.