**Super Keyword-**

It is used to refer immediate parent class variables, method and constructor.

Why?

# **Program-1** for use of immediate parent class variables.

**public** **class** Parent {

**int** x= 20;

}

**class** Child **extends** Parent {

**int** x = 25;

**public** **void** test() {

**int** x= 30;

//Scenario-1

Parent p = **new** Parent();

System.***out***.println("Parent class x variable=" + p. x);

//Scenario-2

System.***out***.println("Immediate super class of child class x variable" + **super**. x);

}

}

**public** **class** TestMain {

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

Child c= **new** Child();

c.test();

}

}

Output-

Parent class x variable=20

Immediate super class of child class x variable20

In Scenario 1, we are calling the x variable of parent class for that purpose we loading the whole class that is not good programmer approach. For use of single variable, we should go for super keyword in java.

In Scenario 2, we are trying to print the x variable of immediate super class by using the super keyword that is the best approach because we are not wasting the memory here.

**# Program 2**- program for use of immediate super class method.

**class** Parent {

**void** test() {

System.***out***.println("Parent class method.");

}

}

**class** Child **extends** Parent {

**void** test() {

**super**.test();

}

}

**public** **class** TestMain {

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

Child c= **new** Child();

c.test();

}

}

Output-

Parent class method.

# **Program-3** to use of immediate super class constructor-

**class** Parent {

Parent() {

System.***out***.println("Parent class constructor.");

}

}

**class** Child **extends** Parent {

Child() {

**super**();

}

}

**public** **class** TestMain {

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

Child c= **new** Child();

}

}

Output-

Parent class constructor.