Q1. Example of Inheritance

class A{

public static void main(String args[]){

System.out.println("This is A class Main");

}

}

class B extends A{

public static void main(String args[]){

System.out.println("This is B class Main");

}

}

Q2. Explain super keyword in java programming?

Ans: super keyword always represent parent class

There are three use of super keyword in java

1. To access Parent class member data from the child class

Syntax:

super.memberDataName

class A{

int y=10;

public A(){

System.out.println("This is A class Default Constructor");

}

public static void main(String args[]){

System.out.println("This is A class Main");

}

}

class B extends A{

int x=20;

public B(){

System.out.println("This is B class Default Constructor");

}

}

class C extends B{

int x=30;

public C(){

System.out.println("This is C class Default Constructor");

}

public void show(){

System.out.println(" This is A Parent X: "+super.y);

System.out.println(" This is B Child x: "+super.x);

System.out.println(" This is C Child x: "+x);

}

public static void main(String args[]){

C obj=new C();

obj.show();

}

}

|  |
| --- |
| This is A class Default Constructor  This is B class Default Constructor  This is C class Default Constructor  This is A Parent X: 10  This is B Child x: 20  This is C Child x: 30 |

1. Using super we call Parent class constructor from the child class

Note: super is a first statement of any constructor or methods

super();//call default constructor of Parent class

super(a,b)

class A{

int x=10;

public A(){

System.out.println("This is A class Default Constructor");

}

public A(int x){

this.x=x;

System.out.println("This is A class Parameterized Constructor");

}

}

class B extends A{

int y=20;

public B(){

System.out.println("This is B class Default Constructor");

}

public B(int x,int y){

super(x);

this.y=y;

System.out.println("This is B class Parameterized Constructor");

}

}

class C extends B{

int z=30;

public C(){

System.out.println("This is C class Default Constructor");

}

public C(int x,int y,int z){

super(x,y);

this.z=z;

System.out.println("This is C class Parameterized Constructor");

}

public void show(){

System.out.println(" This is A Parent X: "+super.x);

System.out.println(" This is B Child x: "+super.y);

System.out.println(" This is C Child x: "+z);

}

public static void main(String args[]){

C obj=new C(1,2,3);

obj.show();

}

}

|  |
| --- |
| This is A class Parameterized Constructor  This is B class Parameterized Constructor  This is C class Parameterized Constructor  This is A Parent X: 1  This is B Child x: 2  This is C Child x: 3 |

1. Super is also used to call Parent class method from the child class

Syntax:

super.methodName();

class A{

int x=10;

public A(){

System.out.println("This is A class Default Constructor");

}

public A(int x){

this.x=x;

System.out.println("This is A class Parameterized Constructor");

}

}

class B extends A{

int y=20;

public B(){

System.out.println("This is B class Default Constructor");

}

public B(int x,int y){

super(x);

this.y=y;

System.out.println("This is B class Parameterized Constructor");

}

public void show(){

System.out.println(" This is A Parent X: "+super.x);

System.out.println(" This is B Child x: "+y);

}

}

class C extends B{

int z=30;

public C(){

System.out.println("This is C class Default Constructor");

}

public C(int x,int y,int z){

super(x,y);

this.z=z;

System.out.println("This is C class Parameterized Constructor");

}

public void show(){

super.show();//to call Parent class method

System.out.println(" This is C Child "+z);

}

public static void main(String args[]){

C obj=new C(1,2,3);

obj.show();

}

}

|  |
| --- |
| This is A class Parameterized Constructor  This is B class Parameterized Constructor  This is C class Parameterized Constructor  This is A Parent X: 1  This is B Child x: 2  This is C Child x: 3 |