DAY-37

------

// EXAMPLE FOR MULTI-LEVEL INHERITANCE

class A

{

void fun1()

{

System.out.println("inside the fun1 method");

}

}

class B extends A

{

void fun2()

{

System.out.println("inside the fun2 method");

}

}

class C extends B

{

void fun3()

{

System.out.println("inside the fun3 method");

}

}

class Demo

{

public static void main(String[] args)

{

C c1 =new C();

c1.fun1();

c1.fun2();

c1.fun3();

}

}

OUTPUT:

-------

inside the fun1 method

inside the fun2 method

inside the fun3 method

// EXAMPLE FOR CYCLIC INHERITANCE

-----------------------------------

class A extends B

{

void fun1()

{

System.out.println("inside the fun1 method");

}

}

class B extends A

{

void fun2()

{

System.out.println("inside the fun2 method");

}

}

class Demo

{

public static void main(String[] args)

{

B b1 =new B();

b1.fun1();

b1.fun2();

}

}

OUTPUT:

--------

error: cyclic inheritance involving A

class A extends B

super keyword

--------------

During the inheritance if the parent class and class having the variable name and the name clash would occur and priority is given to child class

instance variable.

In order to access the instance variable of parent class we will use super keyword.

// EXAMPLE FOR SUPER KEYWORD

class A

{

int x = 10;

}

class B extends A

{

int x=20;

void fun2()

{

System.out.println(x); // 20

System.out.println(this.x); // 20

System.out.println(super.x); // 10

}

}

class Demo

{

public static void main(String[] args)

{

B b1 =new B();

b1.fun2();

}

}

this keyword : it will alwys points the currently executing object.

this() : it is used to call the constructor within the same class.

super() : it is used to call the constructor of the parent class

super keyword : it is used to access the instance variable of parent class.

EXECUTION OF THE INSTANCE MEMBERS IN INHERITANCE:

-------------------------------------------------

EXAMPLE:

--------

class Parent

{

int x = 10;

{

System.out.println("parent class first instance block");

}

{

fun1();

System.out.println("parent class second instance block");

}

Parent()

{

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

}

void fun1()

{

System.out.println("parent class first instance method");

}

}

class Child extends Parent

{

int y=20;

{

System.out.println("child class first instance block");

}

{

fun2();

System.out.println("child class second instance block");

}

Child()

{

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

}

void fun2()

{

System.out.println("child class first instance method");

}

public static void main(String[] args)

{

Child c1= new Child();

}

}

OUTPUT:

--------

parent class first instance block

parent class first instance method

parent class second instance block

parent class constructor

child class first instance block

child class first instance method

child class second instance block

child class constructor

STEPS FOR EXECUTION OF INTANCE MEMBERS IN IN HERITANCE:

-------------------------------------------------------

1. Identify all the instance members fo the parent class

2. execute instance variable fallowed by instance blocks and method and finally execute parent class constrctor

3. Identify all the instance members fo the child class

4. execute instance variable fallowed by instance blocks and method and finally execute child class constrctor

EXECUTION OF THE STATIC MEMBERS IN INHERITANCE:

-------------------------------------------------

EXAMPLE:

--------

class Parent

{

static int x = 10;

static

{

System.out.println("parent class first static block");

}

static

{

fun1();

System.out.println("parent class second static block");

}

Parent()

{

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

}

static void fun1()

{

System.out.println("parent class first static method");

}

}

class Child extends Parent

{

static int y=20;

static

{

System.out.println("child class first static block");

}

static

{

fun2();

System.out.println("child class second static block");

}

Child()

{

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

}

static void fun2()

{

System.out.println("child class first static method");

}

public static void main(String[] args)

{

Child c1= new Child();

}

}

OUTPUT:

-------

parent class first static block

parent class first static method

parent class second static block

child class first static block

child class first static method

child class second static block

parent class constructor

child class constructor

STEPS FOR EXECUTION OF STATIC MEMBERS IN INHERITANCE:

-------------------------------------------------------

1. Identify all the static members fo the parent class

2. execute static variable fallowed by static blocks and method.

3. Identify all the static members fo the child class

4. execute static variable fallowed by static blocks and method.

5. then object is created during object creation constructor will be executed.

EXECUTION OF BOTH STATIC AND INSTANCE MEMBERS IN INHERITANCE:

-------------------------------------------------

EXAMPLE:

--------

class Parent

{

static int x = 10; // static variable

int y = 20; // instance variable

static

{

System.out.println("parent class first static block");

}

// instance block

{

System.out.println("parent class first instance block");

}

// instance block

{

test1();

System.out.println("parent class second instance block");

}

// constructor

Parent()

{

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

}

// static method

static void fun1()

{

System.out.println("parent class first static method");

System.out.println(x);

}

// instance method

void test1()

{

System.out.println("parent class first instance method");

System.out.println(y);

}

}

class Child extends Parent

{

static int a=20;

int b=40;

static

{

System.out.println("child class first static block");

}

{

test2();

System.out.println("child class first instance block");

}

Child()

{

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

}

static void fun2()

{

System.out.println("child class first static method");

System.out.println(a);

}

void test2()

{

System.out.println("child class first instance method");

System.out.println(b);

}

public static void main(String[] args)

{

System.out.println("inside the main method");

Child.fun1();

Child.fun2();

Child c1= new Child();

}

}

OUTPUT:

-------

parent class first static block

child class first static block

inside the main method

parent class first static method

10

child class first static method

20

--------------------------------

parent class first instance block

parent class first instance method

20

parent class second instance block

parent class constructor

child class first instance method

40

child class first instance block

child class constructor

STEPS FOR EXECUTION OF STATIC MEMBERS AND INSTANCE MEMBERS IN INHERITANCE:

---------------------------------------------------------------------------

1. Identify the static members(both parent and child class)

2. execute the sv sb and sm of parent class and execute it in the top-bottom order

3. execute the main method

4. new keyword---> object creation

5. identify the instance members( of both parent and child class)

6. execute the iv ib im and construtor of parent class

7. execute the iv ib im and construtor of child class