DAY-35

--------

INHERITANCE

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

Inheritance is a process of a class aqcuring the properties(variables) and behaviour (methods) from the other class.

Inheritance can be achived using extends keywords.

Parent class is a class which gives the proerties and behaviour to another class.

child class is a class which inherit the proerties and behaviour from its parent class.

ADVANTAGE OF INHERITANCE:

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

1. CODE REUSABLITY.

2. less time for developing the s/w.

example:1

----------

// unrelated classes

class visitor

{

String name;

int ph\_no;

}

class Hospital

{

void Doctor\_conslut()

{

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

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

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

}

}

NOTE:1

-------

In order to relate two classes we can use extends keywords.

example:2

----------

// related classes

class visitor

{

String name; // parent class or super class

int ph\_no;

}

class Hospital extends visitor

{

void Doctor\_conslut() // child class or sub class

{

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

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

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

}

}

NOTE:2

------

Inheritance promotes IS- A relationship.

NOTE:3

------

Whenever the obejct of child class is created the memory is not only allocated for the instance variable of a child class but it also allocates

the instance variable of the parent class.

EXAMPLE:

--------

// note3

class A

{

int i =10; // parent class or super class

}

class B extends A

{

int j = 20;

}

class Demo

{

public static void main(String[] args)

{

B b= new B();

System.out.println(b.i);

System.out.println(b.j);

}

}

NOTE: 4

--------

Private members will not participate in the inheritance.This rule is made to promote ENCAPSULATION.

EXAMPLE:

--------

// note4

class A

{

private int i =10; // parent class or super class

}

class B extends A

{

int j = 20;

}

class Demo

{

public static void main(String[] args)

{

B b= new B();

System.out.println(b.i);

System.out.println(b.j);

}

}

OUTPUT:

--------

i has private access in A

System.out.println(b.i);

NOTE:5

------

Child class can call the instance method of parent class directly.

EXAMPLE:

--------

// note5

class A

{

void fun1()

{

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

}

}

class B extends A

{

void fun2()

{

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

}

}

class Demo

{

public static void main(String[] args)

{

B b= new B();

b.fun2();

b.fun1();

}

}

OUTPUT:

------

inside the fun2

inside the fun1

NOTE:6

------

The child class can call the inherited static methods directly.

EXAMPLE:

--------

// note4

class A

{

static void fun1()

{

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

}

}

class B extends A

{

void fun2()

{

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

}

}

class Demo

{

public static void main(String[] args)

{

B b= new B();

b.fun2();

b.fun1();

}

}

OUTPUT:

------

inside the fun2

inside the fun1

NOTE:7

------

Unlike the private members constructors are also doesnt partcipate in the inheritance.

ie, parent class consructor would not be inherited to the child class rather the control from the child class will go the super class(parent class)

constructor and the code will get executed.

EXAMPLE:

---------

// note7

class A

{

int i;

int j;

A()

{

super();

i=1111;

j=2222;

}

}

class B extends A

{

B()

{

super();

}

void display()

{

System.out.println("i value is "+i);

System.out.println("j value is "+j);

}

}

class Demo

{

public static void main(String[] args)

{

B b= new B();

b.display();

}

}

OUTPUT:

------

i value is 1111

J value is 2222

// note7 example:2

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

class A

{

int i;

int j;

A(int x)

{

super();

i=1111;

j=2222;

}

}

class B extends A

{

B()

{

super();

}

void display()

{

System.out.println("i value is "+i);

System.out.println("j value is "+j);

}

}

class Demo

{

public static void main(String[] args)

{

B b= new B();

b.display();

}

}

OUTPUT:

-------

COMPILATION ERROR

In the child class during the execution of super() a call is made to the default constructor (parent class) if the default constructor is not present

then we will get compilation error

NOTE:8

-------

Even though the parameterized constructor of a sub-class is called yet, the default constructor in the super class is executed first then the parameterzied

constructor of the sub-class is executed.

// note8

-----------

class A

{

int i,j;

A()

{

super();

System.out.println("inside the default constructor");

}

}

class B extends A

{

B()

{

super();

}

B(int i, int j)

{

super();

this.i=i;

this.j=j;

}

void display()

{

System.out.println("i value is "+i);

System.out.println("j value is "+j);

}

}

class Demo

{

public static void main(String[] args)

{

B b= new B(10,20);

b.display();

}

}

OUTPUT:

-------

inside the default constructor

i value is 10

j value is 20

NOTE:9

-------

If the parameterized constructor of the super class has to be executed then parameter super should be used in the sub-class

EXAMPLE:

---------

// note9

class A

{

int i,j;

int a,b;

A(int a,int b)

{

this.a=a;

this.b=b;

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

}

}

class B extends A

{

B(int i, int j)

{

super(30,60);

this.i=i;

this.j=j;

}

void display()

{

System.out.println("i value is "+i);

System.out.println("j value is "+j);

System.out.println("a value is "+a);

System.out.println("b value is "+b);

}

}

class Demo

{

public static void main(String[] args)

{

B b= new B(10,20);

b.display();

}

}

OUTPUT:

-------

inside the parent class constructor

i value is 10

j value is 20

a value is 30

b value is 60