DAY-36

-------

CONSTRUCTOR CHAINING: Calling the constructor of one class from the other class is called as constructor calling.

EXAMPLE:

---------

// CONSTRUCTOR CHAINING:

class A

{

A()

{

super();

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

}

}

class B extends A

{

B()

{

super();

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

}

}

class C extends B

{

C()

{

super();

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

}

}

class Demo

{

public static void main(String[] args)

{

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

C c1= new C();

}

}

OUTPUT:

-------

inside main method

inside the A constructor

inside the B constructor

inside the C constructor

EXAMPLE:

---------

class Account

{

private int acc\_no;

private String eName;

public Account (int acc\_no , String eName)

{

this.acc\_no=acc\_no;

this.eName=eName;

}

public int getAcc\_no()

{

return acc\_no;

}

public String getEname()

{

return eName;

}

}

class SavingAccount extends Account

{

private int balance;

public SavingAccount(int acc\_no,String eName, int balance )

{

super(acc\_no,eName);

this.balance = balance;

}

public int getBalance()

{

return balance;

}

}

class Demo1

{

public static void main(String[] args)

{

SavingAccount sa = new SavingAccount(1234,"subbu",0);

System.out.println(sa.getAcc\_no());

System.out.println(sa.getEname());

System.out.println(sa.getBalance());

}

}

OUTPUT:

--------

1234

subbu

0

TYPES OF INHERITANCE:

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

1. SINGLE INHERITANCE

2. MULTI-LEVEL INHERITANCE

3. HIERARCHICAL INHERITANCE

4. MULTIPLE INHERITANCE

5. HYBRID INHERITANCE

6. CYCLIC INHERITANCE

refer the digram:

NOTE: In Java single,multi-level and hierarchical inheritance is supported and other types are not supported.

Why in java multiple inheritance is not supported?

--> In case of multiple inheritance it will result in ambiguity(confusion).This problem is referred as 'DIAMOND SHAPE' problem.

EXAMPLE:

--------

class A

{

int i = 10;

}

class B extends A

{

int i =20;

}

class C extends A,B

{

void display()

{

System.out.println(i);

}

}

class Demo2

{

public static void main(String[] args)

{

C c1 = new C();

c1.display();

}

}

OUTPUT:

--------

COMPILATION ERROR