**Day 14**

**Multiple Inheritance & Hybrid Inheritance not supported in Java**

class A

{

void print()

{

System.out.rpintln(“Print”);

}

}

class B

{

void print()

{

System.out.rpintln(“Printing”);

}

}

class C extends A, B

{

C ob = new C();

ob.print(); // there will be confusion

}

* **An ambiguity is generated in multiple inheritance. So, it is not supported in Java.**
* **Hybrid inheritance is a part of multiple inheritance. So, it is also not supported.**

**Super Keyword**

* It is a reference variable.
* It is used to invoke immediate parent class object.
* It has three usages. They are:

1. It is used to refer immediate parent class instance variable.
2. It is used to invoke immediate parent class method.
3. It is used to invoke immediate parent class constructor.

Referring immediate parent class instance variable.

class A

{

String color=”Black”;

}

class B extends A

{

String color =”White”;

void print()

{

//System.out.println(color);

System.out.println(super.color);

}

class Test

{

public static void main(String args)

{

B ob = new B();

ob.print();

}

Output:

Black

Invoke parent class method

class A

{

Void show()

{

System.out.println(“Show”);

}

class B extends A

{

void show()

{

System.out.println(“Display”);

}

void print()

{

System.out.println(“Print”);

}

void display()

{

Super.show();

print();

}

}

class Test

{

public static void main(String args)

{

B ob = new B();

ob.display();

}

Output:

Show

Print

Invoke parent class Constructor

class A

{

A()

{

System.out.println(“Default constructor”);

}

class B extends A

{

B()

{

super();

System.out.println(“Constructor”);

}

}

class Test

{

public static void main(String args)

{

B ob = new B();

}

Output:

Default Constructor

Constructor