

Output

default constructor is invoked

default constructor is invoked

111 Karan

222 Aryan

Why multiple inheritance is not supported in java

20

```
class A
```

```
{
```

```
    void msg()
```

```
    { System.out.println("Hello");
```

```
    }
```

```
}
```

```
class B
```

```
{
```

```
    Void msg()
```

```
    {
```

```
        System.out.println("Welcome");
```

```
    }
```

```
}
```

```
class C extends A,B
```

```
{
```

```
    Public Static void main(String args[])
```

```
    {
```

```
        C obj=new C();
```

```
        obj.msg();
```

```
    }
```

```
}
```

By : Jigar M Pandya

Preview from Notesale.co.uk
Page 20 of 34

Abstraction

Preview from Notesale.co.uk
Page 21 of 34

■ **Abstraction** is a process of hiding the implementation details and showing only functionality to the user

Ways to achieve Abstraction

- There are two ways to achieve abstraction in java
 1. Abstract class
 2. Interface

Interface in Java

- An **interface in java** is a blueprint of a class.
- It has static constants and abstract methods only.
- The interface in java is **a mechanism to achieve fully abstraction.**
- There can be only abstract methods in the java interface not method body.
- It is used to achieve fully abstraction and multiple inheritance in Java.

Multiple inheritance is not supported through class in java but it is possible by interface, why?

```
interface Printable
```

```
{  
    void print();  
}
```

```
interface Showable
```

```
{  
    void print();  
}
```

```
class TestInterface1 implements Printable, Showable  
{
```

```
    public void print(){System.out.println("Hello");}
```

```
    public static void main(String args[])
```

```
{
```

```
        TestInterface1 obj = new TestInterface1();
```

```
        obj.print();
```

```
    }
```

```
}
```

By : Jigar M Pandya