Interface: Multiple Inheritance

Interface is like a class but includes a group of method declarations (only abstract methods) and final fields.This means that interface do not specify any code to implement these methods and data field contain only constants.Therefore it is responsibility of a class that implements an interface to define the code for implementation of these methods.

Syntax

interface interfacename

{

Variable declaration;

Methods declaration;

}

Where interface is a keyword and interfacename is any valid name given to it.

The variables inside interface are declared as

static final datatype variable name = value;

e.g. static final float pi=3.14;

and the methods are declared as

returntype of method method name (parameter list);

e.g

interface test

{

static final float pi=3.14;

Void showpi();

}

Or

interface area

{

final float pi=3.14;

float area(float r);

}

Interfaces can be extended from other interfaces

How to implement an interface

Class classname implements interfacename

{

//Body of class

}

Interface can be implemented by the use of a keyword implements we can also implement multiple interfaces

e.g.

```
interface area
{
final  float pi=3.14;
float area(float r);
}
class  areacircle implements area
{
public  float area(float r)
{
return(pi*r*r);
}
class test
{
public static void main(String args[])
{
areacircle c= new areacircle();
System,out.println(" Area of circle with radius 2 is "+ c.area(2));
}
}
```

Accessing Interface variables

Interface variables can be accessed in a class which implements that interface

e.g. interface A

{

final int X=100;

}

class B implements A

{

int y =X;

// methods

//statements

}

Program to get multiple inheritance using interface

```
import java.io.*;
class A
{
int rno;
void getrno(int z)
{rno=z;
}
void putrno()
{
System.out.println(" Roll No "+rno);
}
}
class B extends A
{
float m1,m2;
void getmarks(float a,float b)
{
m1=a;
m2=b;
}
```

```
void putmarks()
{
System.out.println(" The marks are ");
System.out.println(" Marks1 "+m1);
System.out.println(" Marks 2"+m2);
}
}

interface C
{
flaot sportmarks = 5.0;
void putsports();
}

class D extends B implements C
{
float total;
public void putsports()
{
System.out.println(" Sprots marks are "+sportmarks);
}

void show()
{
total = m1+m2+sportmarks;
putrno();
putmarks();
putsprots();
System.out.println(" Total is "+total);
}

}

class hybrid
{
public static void main(String args[])
{

D d=new D();
d.getrno(10);
d.getmarks(78,88);
d.show();
}

}
```