**Practical No.8**

**Name**: Anuj Rajendra Mane

**Div**: SY A

**Roll No**: 65 **Batch**: A3

**PRN** **No**: 2162701242054

**Title: Program on Inheritance and interface**

**Inheritance:**

1. The process of getting property of one class into another class is called Inheritance.
2. In other word we can say that the process of deriving a new class from an old class is called inheritance in which the new class is called derived or child or sub class and old class is called Base or Parent or Super class.
3. When a class inherits the property of a class it means it can access all the data member and member function of that class except private element.
4. In this type of programming mainly two types of classes are used.

**Parent/Super/Base class**

**Child/Sub/Derived class**

* **Parent/Super/Base class**

The class which is inherited by another class is called Parent or Super or Base class.

* **Child/Sub/Derived class**

The class which inherits the property of another class is called Child or Sub or Derived class.

**How to inherit one class into another**

Derived class extends Base class

Example

class Subtraction extends Addition

Here Subtraction is a Derived class and Addition is a Base class and extends is a keyword which is used to inherit one class into another.

**Types of Inheritance:**

Single Inheritance

Multiple Inheritance

Multilevel Inheritance

Hierarchical Inheritance

Hybrid Inheritance

**1: Single Inheritance**

In this types of inheritance only two classes are used in which one is inherited by another.

**Program**:

//Base class

class Addition

{

void add()

{

int x,y=30,z=10;

x=y+z;

System.out.println("Add="+x);

}

}

//Derived class extending base class

class Subtraction extends Addition

{

void sub()

{

int x,y=30,z=10;

x=y-z;

System.out.println("Sub="+x);

}

}

class Easy

{

public static void main(String[] args)

{

//Creating instance(object)

Subtraction obj=new Subtraction();

//calling base class method

obj.add();

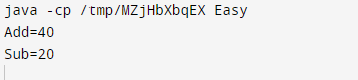
//calling derived class method

obj.sub();

}

}

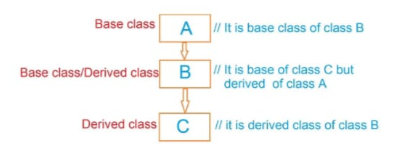
**Output**:



**2. Multilevel Inheritance**

1.When first class is inherited by second class, second class is inherited by third class and inheritance. SO on called multilevel

2.In this type of inheritance each derived class is the base class for the next class.

3.In this type of inheritance at least three class are compulsory.

class Addition

{

public void add()

{

int x,y=30,z=10;

x=y+z;

System.out.println("Add="+x);

}

}

//extending Addition

class Subtraction extends Addition

{

void sub()

{

int x,y=30,z=10;

x=y-z;

System.out.println("Sub="+x);

}

}

//extending Subtraction

class Multiplication extends Subtraction

{

void multi()

{

int x,y=30,z=10;

x=y\*z;

System.out.println("Multiply="+x);

}

}

class Easy

{

public static void main(String[] args)

{

//Creating instance(object)

Multiplication obj=new Multiplication();

obj.add();

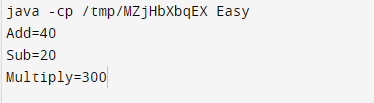
obj.sub();

obj.multi();

}

}

**Output**:



**3. Multiple Inheritance**

1. When two or more than two classes are inherited by a single class simultaneously called multiple inheritance.

2.In other word we can say that in this type of inheritance Base class may be two or more than two but derived class should be one.

3.In this type of inheritance at least three class are compulsory.

4.Java does not support multiple inheritance therefor interface are used to implement multiple inheritance.

5.interface is declared with interface keyword and it is implemented by a class while class is extended by a class.

6.we cannot define function inside an interface, only can be declared.

7.It is the responsibility of derive class to implement/define the method of interface.

**Program:**

//interface

interface Addition

{

//declaring method

//because we can not define function inside interface

void add();

}

//Derived class

class Subtraction

{

void sub()

{

int x,y=30,z=10;

x=y-z;

System.out.println("Sub="+x);

}

}

//Derived class extending base class

//and implementing interface

class Multiplication extends Subtraction implements Addition

{

//implementing method of interface

public void add()

{

int x,y=30,z=10;

x=y+z;

System.out.println("Add="+x);

}

void multi()

{

int x,y=30,z=10;

x=y\*z;

System.out.println("Multiply="+x);

}

}

class Easy

{

public static void main(String[] args)

{

//Creating instance(object)

Multiplication obj=new Multiplication();

obj.add();

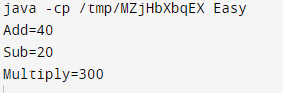
obj.sub();

obj.multi();

}

}

Output:



1. **Hierarchical Inheritance**

1.When a single class is inherited by two or more than two classes simultaneously called hierarchical inheritance.

2.In other word we can say that in this type of inheritance derived class may be two or more than two but Base class should be one.

3.In this type of inheritance at least three class are compulsory.

**Program:**

class Addition

{

public void add()

{

int x,y=30,z=10;

x=y+z;

System.out.println("Add="+x);

}

}

//extending Addition

class Subtraction extends Addition

{

void sub()

{

int x,y=30,z=10;

x=y-z;

System.out.println("Sub="+x);

}

}

//extending same class Addition

class Multiplication extends Addition

{

void multi()

{

int x,y=30,z=10;

x=y\*z;

System.out.println("Multiply="+x);

}

}

class Easy

{

public static void main(String[] args)

{

//Creating instance(object)

Multiplication obj=new Multiplication();

//calling base class function

obj.add();

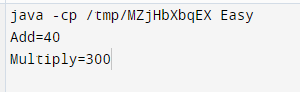
//calling derive class function

obj.multi();

}

}

Output:



**Interfaces:**

1.interface is an important element in java which is called reference type.

2.It is blueprint of class.

3.It is a collection of abstract method/function.

4.It is declared with interface keyword.

5.We cannot create instance(object) of an interface.

6.We cannot define function inside interface only can be declared, so it is the responsibility of derived class to implement the method/function of interface.

7.A class is extended by a class but an interface is implemented by a class.

**Program**:

interface Geometry

{

void rectangle\_area(int height,int width);

void square\_area(int side);

void circle\_area(float radius);

}

//implementing interface

class Easy implements Geometry

{

//implementing method of interface

public void rectangle\_area(int height,int width)

{

int ar=height\*width;

System.out.println("Area of rectangle="+ar);

}

//implementing method of interface

public void square\_area(int side)

{

int ar=side\*side;

System.out.println("Area of square="+ar);

}

//implementing method of interface

public void circle\_area(float radius)

{

float ar=3.14f\*radius\*radius;

System.out.println("Area of circle="+ar);

}

public static void main(String[] args)

{

//creating instance of derived class

Easy obj=new Easy();

obj.rectangle\_area(12, 13);

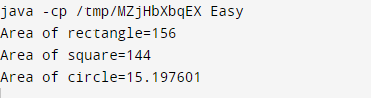
obj.square\_area(12);

obj.circle\_area(2.2f);

}

}

**Output**:



**Conclusion**:

I can learn the basic concepts of inheritance in this concept there are five types I can implementing this all five types as well as understanding the concept deeply or also doing the concept of Interfaces.

**Completion Date: Co-Ordinator Sign:**