**Concept of Inheritance**

Inheritance is one of Important concept of Object-oriented Programming. It is mechanism in java in which one property can inherit feutures and method of another class.

In Java, Inheritance means creating new class based on the existing one.

A class that inherits from another class can reuse the methods an d fields of that class.

NEED of Inheritance.

For achieving Abstraction, Method Overriding, code reusability.

We can achieve Single, Multilevel, hierarchical, multiple (through Interface) Inheritance

1. SINGLE INHERITANCE

class Bicycle {

// the Bicycle class has two fields

public int gear;

public int speed;

// the Bicycle class has one constructor

public Bicycle(int gear, int speed)

{

this.gear = gear;

this.speed = speed;

}

// the Bicycle class has three methods

public void applyBrake(int decrement)

{

speed -= decrement;

}

public void speedUp(int increment)

{

speed += increment;

}

// toString() method to print info of Bicycle

public String toString()

{

return ("No of gears are " + gear + "\n"

+ "speed of bicycle is " + speed);

}

}

// derived class

class MountainBike extends Bicycle {

// the MountainBike subclass adds one more field

public int seatHeight;

// the MountainBike subclass has one constructor

public MountainBike(int gear, int speed,

int startHeight)

{

// invoking base-class(Bicycle) constructor

super(gear, speed);

seatHeight = startHeight;

}

// the MountainBike subclass adds one more method

public void setHeight(int newValue)

{

seatHeight = newValue;

}

// overriding toString() method

// of Bicycle to print more info

@Override public String toString()

{

return (super.toString() + "\nseat height is "

+ seatHeight);

}

}

// driver class

public class Test {

public static void main(String args[])

{

MountainBike mb = new MountainBike(3, 100, 25);

System.out.println(mb.toString());

}

}

Output

No of gears are 3

speed of bicycle is 100

seat height is 25

refer : <https://www.geeksforgeeks.org/inheritance-in-java/>

1. MULTIPLE INHERITANCE

public interface InterFaceOne{

public void one();

}

public interface InterFaceTwo{

public void two();

}

public interface InterFaceThree extends InterFaceOne,InterFaceTwo{

public void three();

}

public class MultipleClass implements InterFaceThree{

@Override

public void one()

{

System.out.println("Interface One");

}

@Override

public void two()

{

System.out.println("Interface Two");

}

@Override

public void three()

{

System.out.println("Interface Three");

}

}

public class Main

{

public static void main (String[]args)

{

MultipleClass mp = new MultipleClass();

mp.one();

mp.two();

mp.three();

}

}

Output

Interface One

Interface Two

Interface Three