Experiment No 8

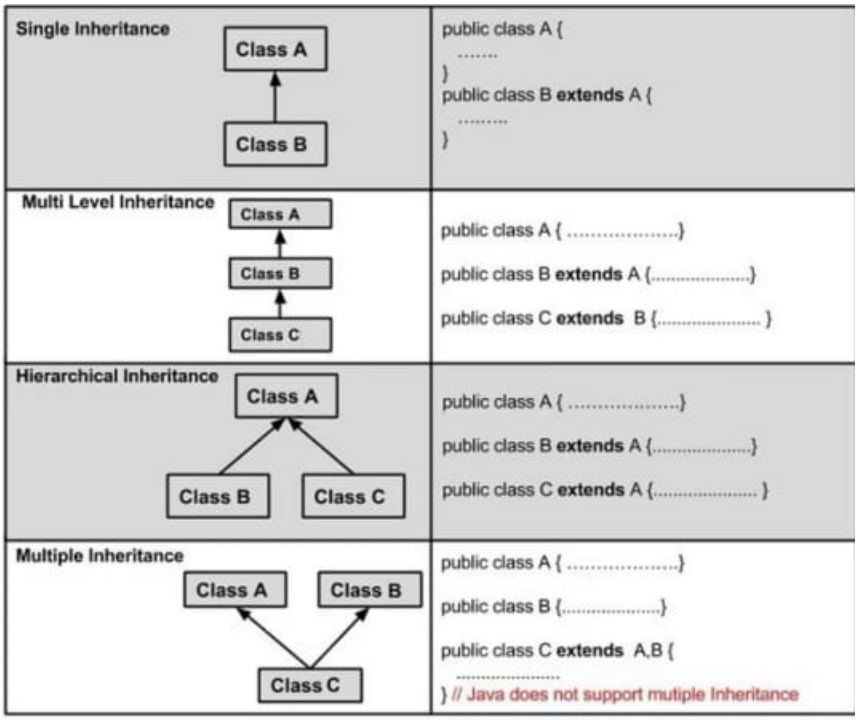
Aim: To implement Programs on types of inheritance.

Theory:

Inheritance can be defined as the process where one class acquires the properties (methods and fields) of another. With the use of inheritance the information is made manageable in a hierarchical order.

The class which inherits the properties of others is known as subclass (derived class, child class) and the class whose properties are inherited is known as superclass (base class, parent class).extends Keyword extends is the keyword used to inherit the properties of a class. Below given is the syntax of extends keyword.

```
class Super1{

.....

.....

}

class Sub extends Super1{

.....

.....

}
```



Programs:

//An Example of Inheritance- Single

```
class Bicycle {
    // the Bicycle class has three fields
    public int pedalling_Rate;
    public int gear;
    public int speed;
   // the Bicycle class has one constructor
    public Bicycle(int start_Pedalling_Rate, int startSpeed, int startGear) {
        gear = startGear;
        pedalling_Rate = start_Pedalling_Rate;
        speed = startSpeed;
    }
 // the Bicycle class has four methods
```

```
    public void set_Pedalling_Rate(int newValue) {
        pedalling_Rate = newValue;
    }
    public void setGear(int newValue) {
        gear = newValue;
    }
    public void applyBrake(int decrement) {
        speed -= decrement;
    }
 public void speedUp(int increment) {
        speed += increment;
    }}
public class MountainBike_Inheritance extends Bicycle {
    // the MountainBike_Inheritance subclass adds one field
    public int seatHeight;
  // the MountainBike_Inheritance subclass has one constructor
    public MountainBike_Inheritance(int startHeight,
                        int start_Pedalling_Rate,
                        int startSpeed,
                        int startGear) {
        super(start_Pedalling_Rate, startSpeed, startGear);
        seatHeight = startHeight;
    }
    // the MountainBike_Inheritance subclass adds one method
    public void setHeight(int newValue) {
```

```
        seatHeight = newValue;
    }
public static void main(String []args)
        {
                MountainBike_Inheritance bike= new MountainBike_Inheritance(4,60,10,3);
        System.out.println("The start height is "+bike.seatHeight+ "\n Start Pedalling rate is "
+bike.pedalling_Rate+ "RPM.\n Start speed is "+bike.speed+ "km/hr \n Start Gear is "
+bike.gear);


        //setting new values for Bicycle in motion
        bike.set_Pedalling_Rate(80);
        bike.setGear(5);
        bike.applyBrake(2);
        bike.speedUp(2);
        bike.setHeight(4);
        System.out.println("When in motion the height is "+bike.seatHeight+ "\n Start Pedalling
rate is " +bike.pedalling_Rate+ "RPM.\n Start speed is "+bike.speed+ " km/hr \n Start Gear is "
+bike.gear);
        }
}
//An Example of Inheritance- Hierarchical
    class Animal{
    void eat(){System.out.println("eating...");}
    }
    class Dog extends Animal{
    void bark(){System.out.println("barking...");}
    }
```

```
class Cat extends Animal{
void meow(){System.out.println("meowing...");}
}
class TestInheritance3{
public static void main(String args[]){
Cat c=new Cat();
c.meow();
c.eat();
//c.bark();//C.T.Error
}}
```

Exercise:

1. Create a class named furniture having dimensions as l,b and h. Inherit the class to create a child class named chair(having data member as number of legs).
2. Implement following multilevel inheritance. Take data members and member functions as per your choice. Student←Marks←Grade
3. Create a furniture class as example no1. Inherit one more class shelf from furniture (having data member as no of compartments) to demonstrate hierarchical inheritance.

Conclusion: In this experiment we see the types of inheritance and the use of extends keyword. Inheritance avoids code reuse thus increasing its efficiency.