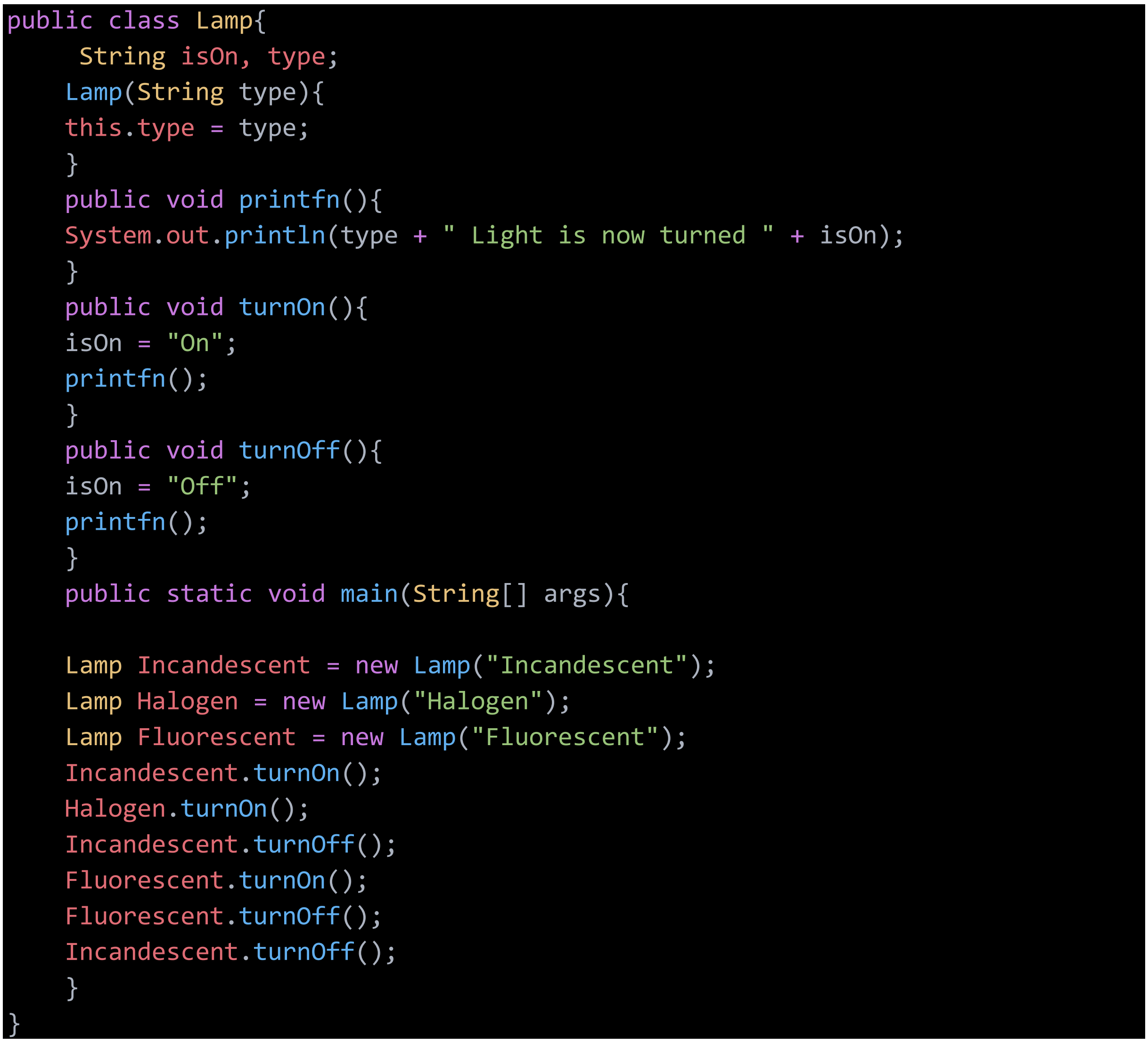
JAVA Week 4 Assignment

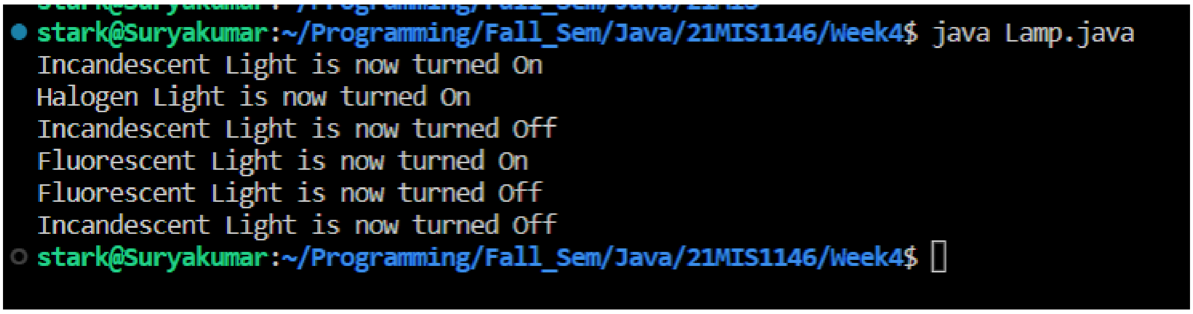
Suryakumar P 21MIS1146

1)



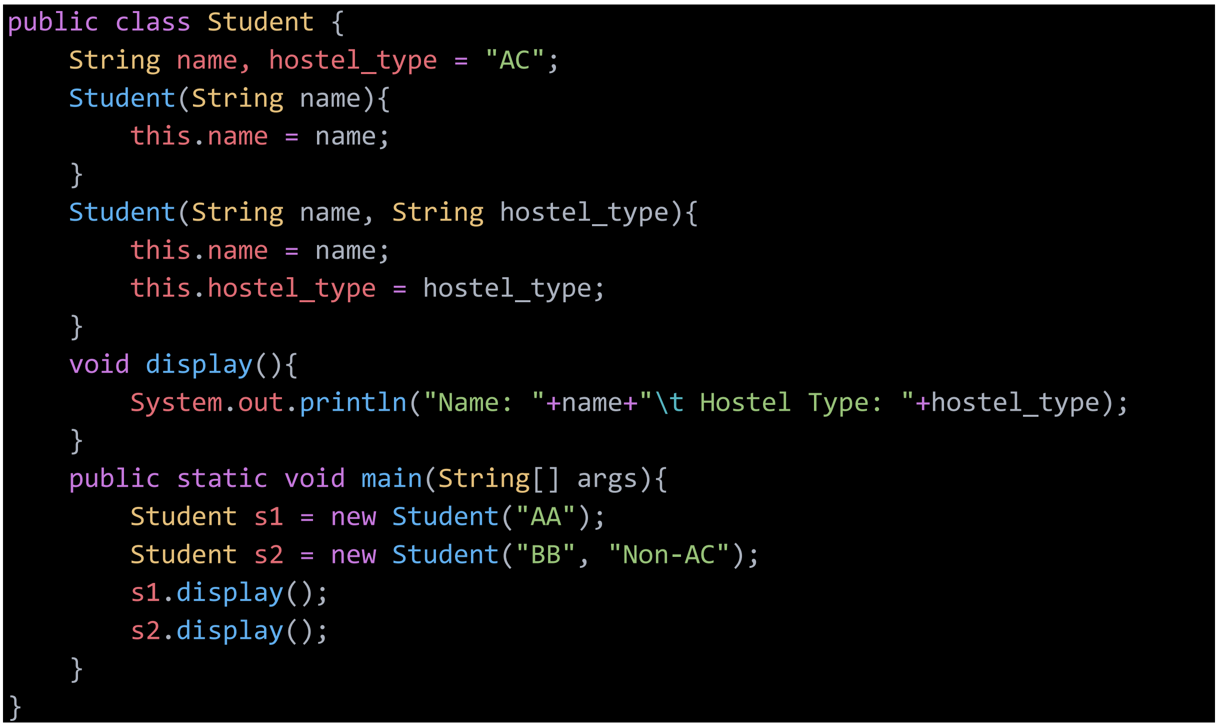
```java
public class Lamp{
     String isOn, type;
    Lamp(String type){
    this.type = type;
    }
    public void printfn(){
    System.out.println(type + " Light is now turned " + isOn);
    }
    public void turnOn(){
    isOn = "On";
    printfn();
    }
    public void turnOff(){
    isOn = "Off";
    printfn();
    }
    public static void main(String[] args){

    Lamp Incandescent = new Lamp("Incandescent");
    Lamp Halogen = new Lamp("Halogen");
    Lamp Fluorescent = new Lamp("Fluorescent");
    Incandescent.turnOn();
    Halogen.turnOn();
    Incandescent.turnOff();
    Fluorescent.turnOn();
    Fluorescent.turnOff();
    Incandescent.turnOff();
    }
}
```

Output:



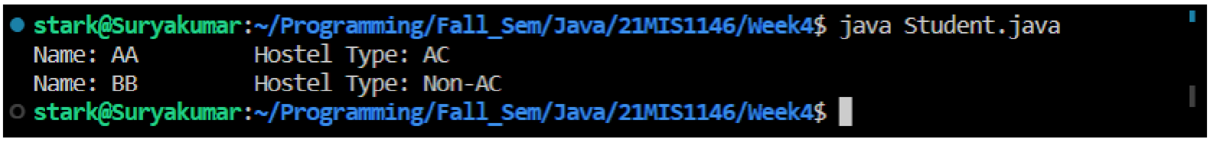
```
  stark@Suryakumar:~/Programming/Fall_Sem/Java/21MIS
● stark@Suryakumar:~/Programming/Fall_Sem/Java/21MIS1146/Week4$ java Lamp.java
  Incandescent Light is now turned On
  Halogen Light is now turned On
  Incandescent Light is now turned Off
  Fluorescent Light is now turned On
  Fluorescent Light is now turned Off
  Incandescent Light is now turned Off
○ stark@Suryakumar:~/Programming/Fall_Sem/Java/21MIS1146/Week4$ ▯
```

2)



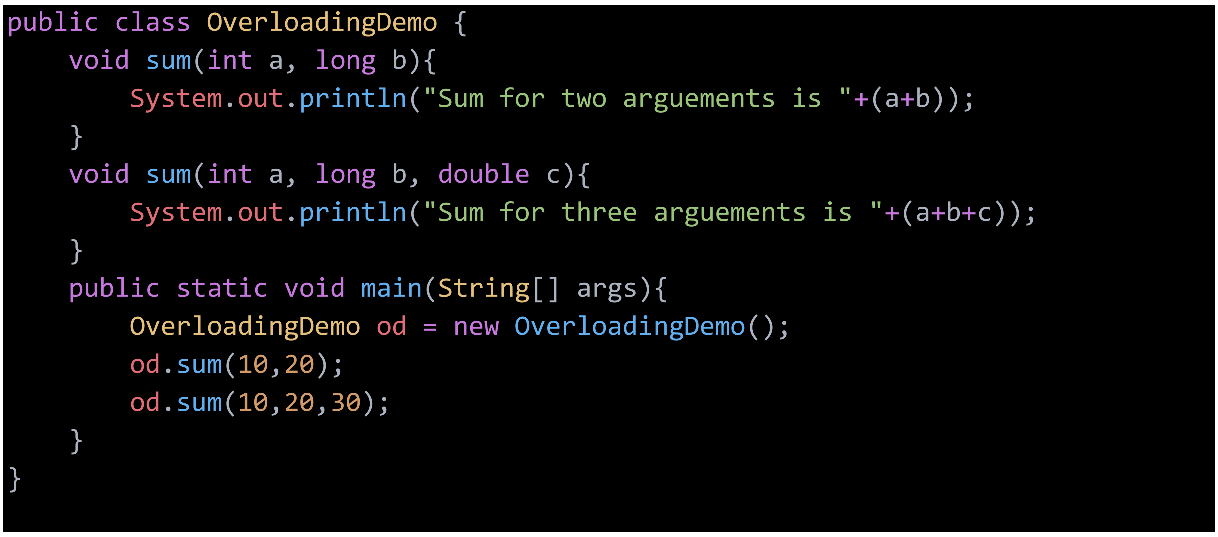
```java
public class Student {
    String name, hostel_type = "AC";
    Student(String name){
        this.name = name;
    }
    Student(String name, String hostel_type){
        this.name = name;
        this.hostel_type = hostel_type;
    }
    void display(){
        System.out.println("Name: "+name+"\t Hostel Type: "+hostel_type);
    }
    public static void main(String[] args){
        Student s1 = new Student("AA");
        Student s2 = new Student("BB", "Non-AC");
        s1.display();
        s2.display();
    }
}
```

Output:



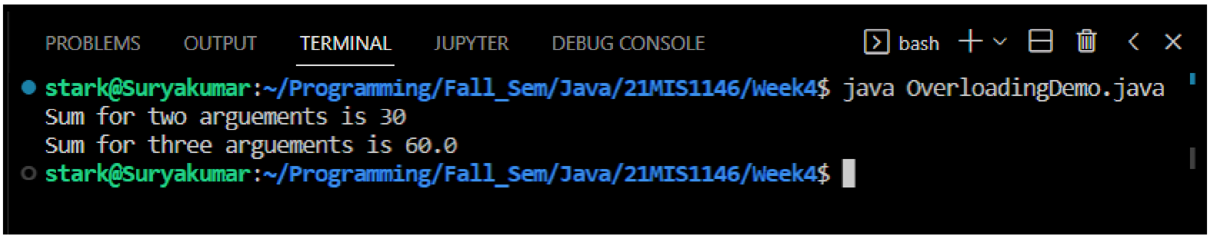
```
stark@Suryakumar:~/Programming/Fall_Sem/Java/21MIS1146/Week4$ java Student.java
Name: AA         Hostel Type: AC
Name: BB         Hostel Type: Non-AC
stark@Suryakumar:~/Programming/Fall_Sem/Java/21MIS1146/Week4$ 
```

3)



```java
public class OverloadingDemo {
    void sum(int a, long b){
        System.out.println("Sum for two arguements is "+(a+b));
    }
    void sum(int a, long b, double c){
        System.out.println("Sum for three arguements is "+(a+b+c));
    }
    public static void main(String[] args){
        OverloadingDemo od = new OverloadingDemo();
        od.sum(10,20);
        od.sum(10,20,30);
    }
}
```

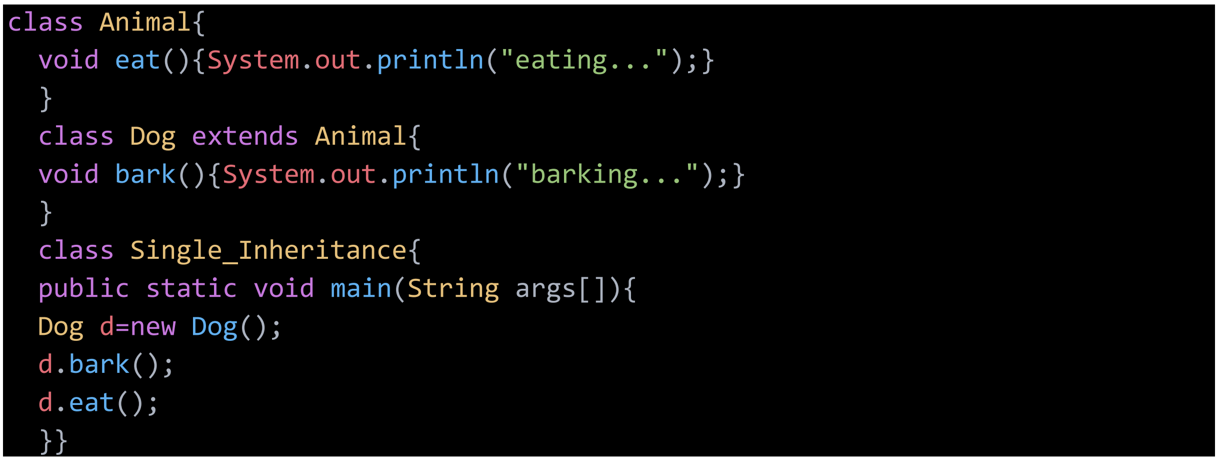
Output:



```
stark@Suryakumar:~/Programming/Fall_Sem/Java/21MIS1146/Week4$ java OverloadingDemo.java
Sum for two arguements is 30
Sum for three arguements is 60.0
stark@Suryakumar:~/Programming/Fall_Sem/Java/21MIS1146/Week4$ 
```

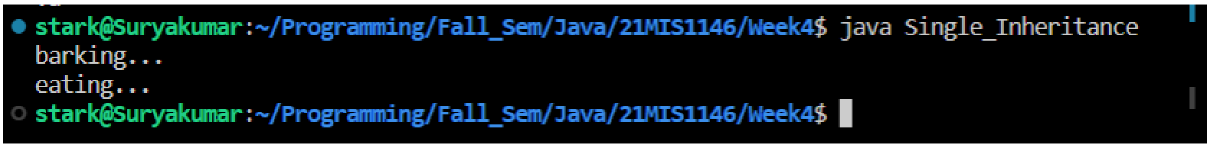
4)

Single:



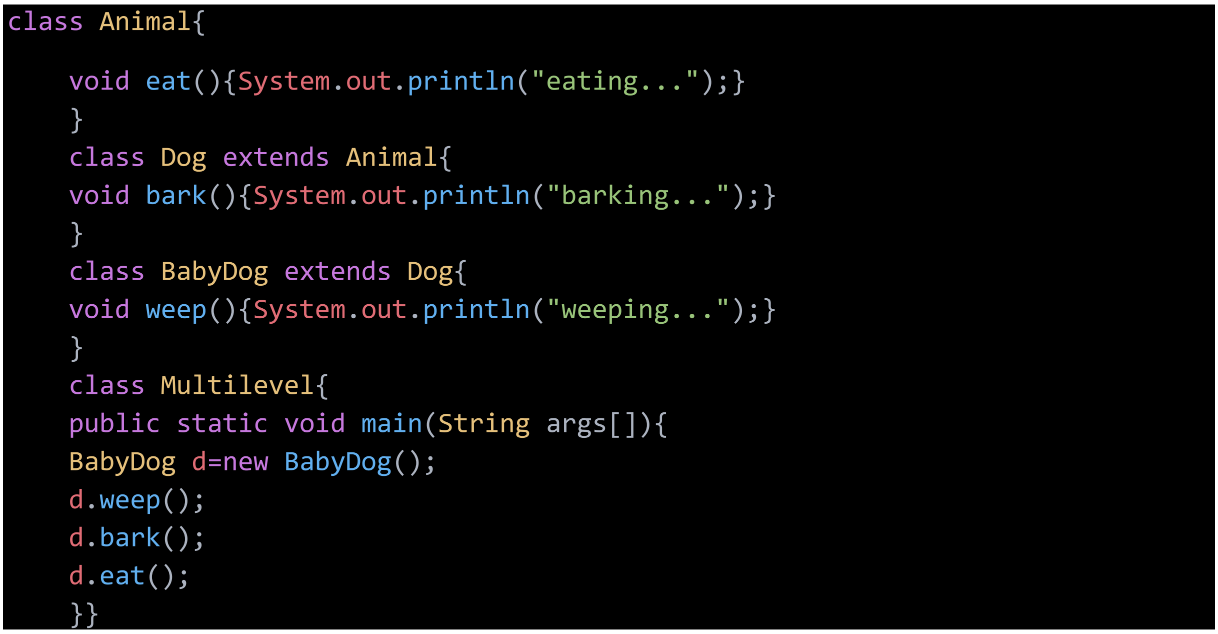
```
class Animal{
  void eat(){System.out.println("eating...");}
  }
  class Dog extends Animal{
  void bark(){System.out.println("barking...");}
  }
  class Single_Inheritance{
  public static void main(String args[]){
  Dog d=new Dog();
  d.bark();
  d.eat();
  }}
```

Output:



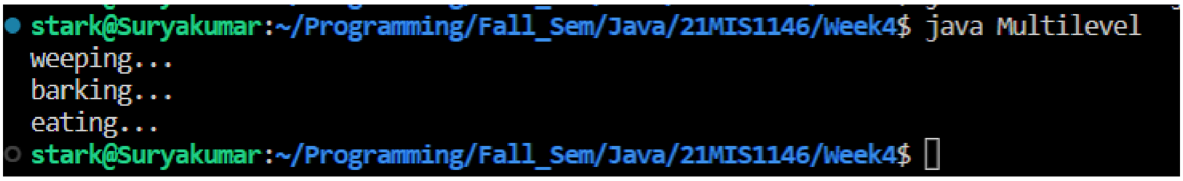
```
● stark@Suryakumar:~/Programming/Fall_Sem/Java/21MIS1146/Week4$ java Single_Inheritance
  barking...
  eating...
○ stark@Suryakumar:~/Programming/Fall_Sem/Java/21MIS1146/Week4$
```

Multilevel:



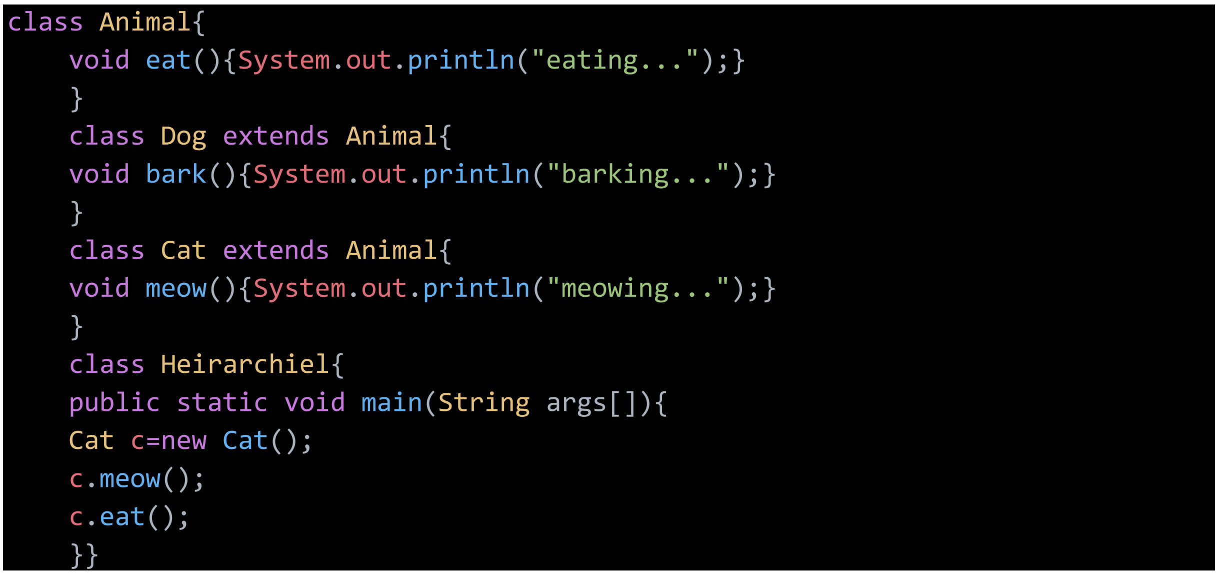
```
class Animal{

    void eat(){System.out.println("eating...");}
    }
    class Dog extends Animal{
    void bark(){System.out.println("barking...");}
    }
    class BabyDog extends Dog{
    void weep(){System.out.println("weeping...");}
    }
    class Multilevel{
    public static void main(String args[]){
    BabyDog d=new BabyDog();
    d.weep();
    d.bark();
    d.eat();
    }}
```

Output:



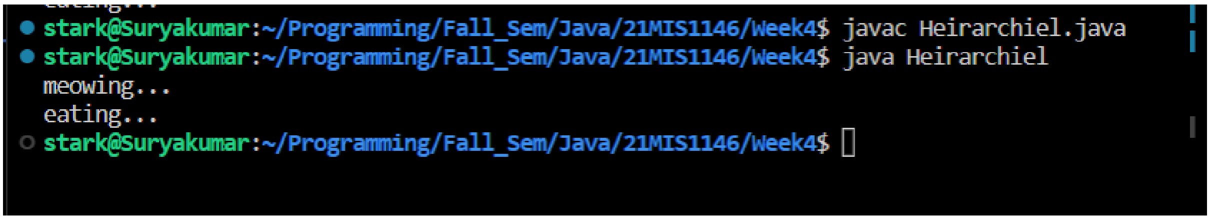
```
● stark@Suryakumar:~/Programming/Fall_Sem/Java/21MIS1146/Week4$ java Multilevel
  weeping...
  barking...
  eating...
○ stark@Suryakumar:~/Programming/Fall_Sem/Java/21MIS1146/Week4$
```

Heirarchiel:



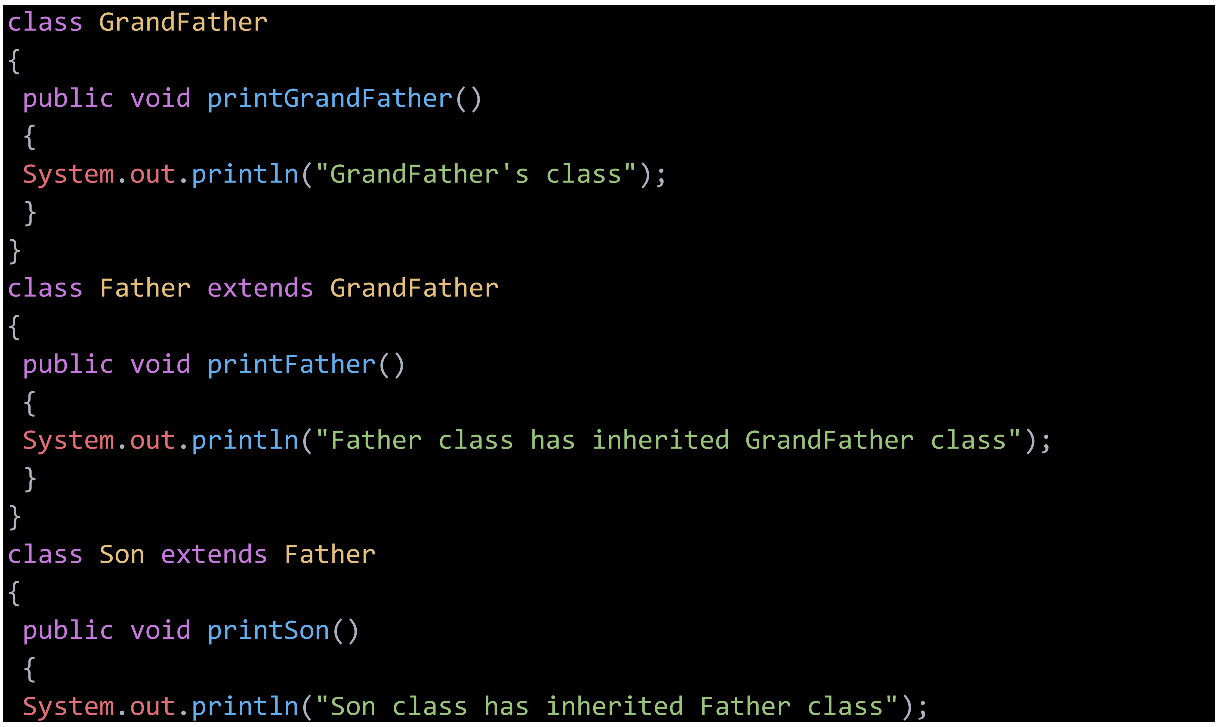
```
class Animal{
    void eat(){System.out.println("eating...");}
    }
    class Dog extends Animal{
    void bark(){System.out.println("barking...");}
    }
    class Cat extends Animal{
    void meow(){System.out.println("meowing...");}
    }
    class Heirarchiel{
    public static void main(String args[]){
    Cat c=new Cat();
    c.meow();
    c.eat();
    }}
```

Output:

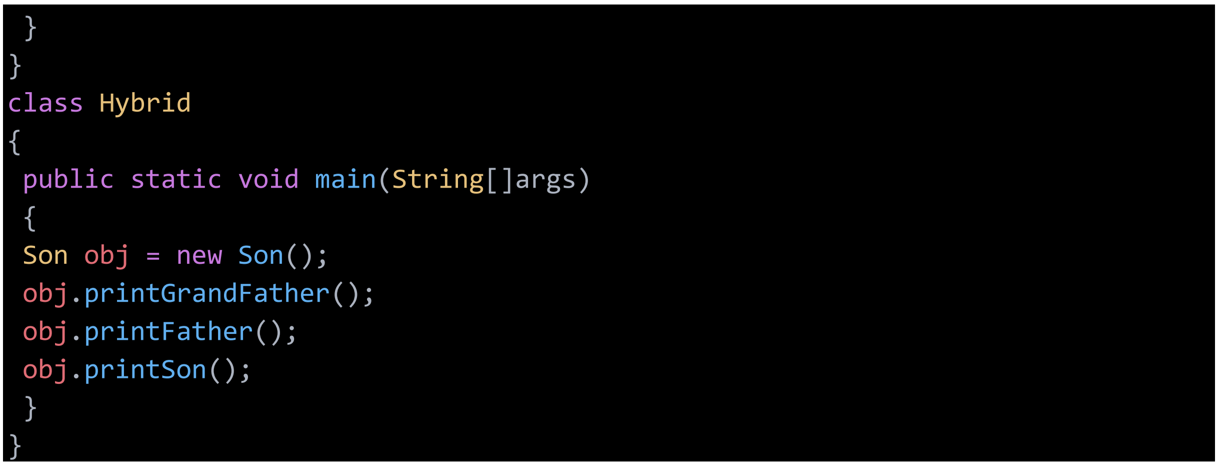


```
  eating...
● stark@Suryakumar:~/Programming/Fall_Sem/Java/21MIS1146/Week4$ javac Heirarchiel.java
● stark@Suryakumar:~/Programming/Fall_Sem/Java/21MIS1146/Week4$ java Heirarchiel
  meowing...
  eating...
○ stark@Suryakumar:~/Programming/Fall_Sem/Java/21MIS1146/Week4$ 
```

Hybrid:

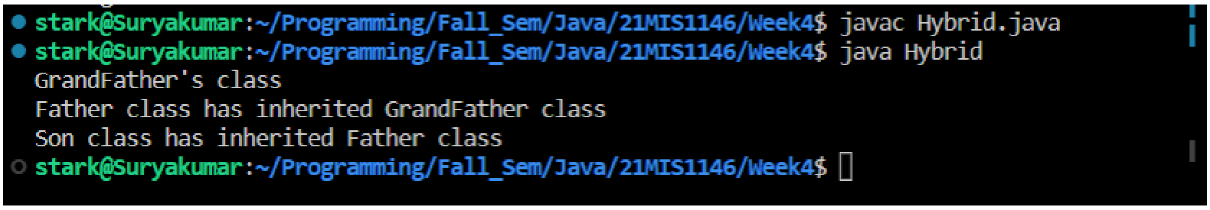


```
class GrandFather
{
 public void printGrandFather()
 {
 System.out.println("GrandFather's class");
 }
}
class Father extends GrandFather
{
 public void printFather()
 {
 System.out.println("Father class has inherited GrandFather class");
 }
}
class Son extends Father
{
 public void printSon()
 {
 System.out.println("Son class has inherited Father class");
```



```
 }
}
class Hybrid
{
 public static void main(String[]args)
 {
 Son obj = new Son();
 obj.printGrandFather();
 obj.printFather();
 obj.printSon();
 }
}
```

Output:



```
stark@Suryakumar:~/Programming/Fall_Sem/Java/21MIS1146/Week4$ javac Hybrid.java
stark@Suryakumar:~/Programming/Fall_Sem/Java/21MIS1146/Week4$ java Hybrid
GrandFather's class
Father class has inherited GrandFather class
Son class has inherited Father class
stark@Suryakumar:~/Programming/Fall_Sem/Java/21MIS1146/Week4$
```

Multiple:

Java does not support multiple inheritance