Assignment

JAVA

C.K. Afroz
192311108
CSA0985

INHERITANCE:

Inheritance means creating new classes faced on existing ones. Inheritance in Java is a key feature of object-oriented programming that allows one class to inherit the properties and behaviours of another class.

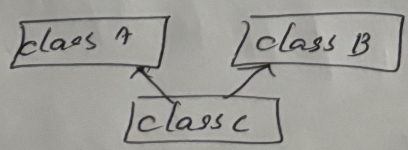
Types of Inheritance:

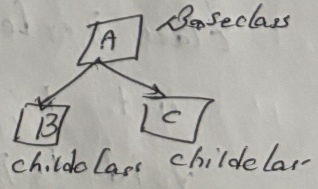
1. Single Inheritance: A class inherits from one Superclass

class A → class B
(Parent) (Child)

2. Multilevel Inheritance: A class is derived from a class, which is also derived from another class

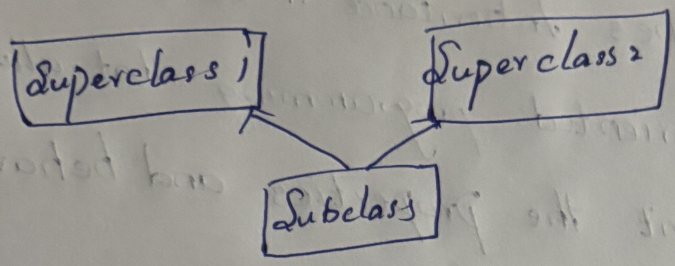
3. Hierarchial Inheritance: Multiple classes inherit from a Single Superclass



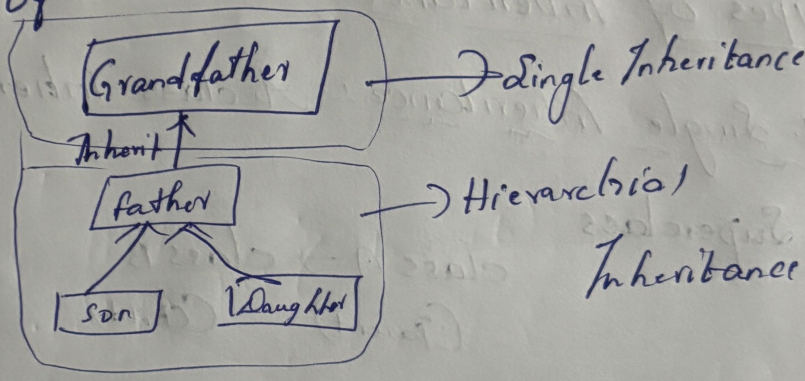


4) Multiple Inheritance:-

A Scenario where a class can inherit properties and methods from more than one super class



5. Hybrid Inheritance:- It is a combination of two or more types of inheritance



Single Inheritance:-

```
class A{
  int a;
  Void display A(){
    System.out.println("a = "+a);
  }
}
class B extends A{
  int b;
  void display B(){
```

```
System.out.println("b="+b);
}
}
Public class Single Inheritance Example{
    Public static Void main(String[] args){
```

~~Output:~~

```
B obj=new B();
obj.a=20;
obj.b=30;
obj.displayA();
obj.displayB();
}
}
```

Output: a=20, b=30

Multilevel Inheritance:

```
class A{
    Public Void displayA(){
    System.out.println("Inside display A");
    }
}
class B extends A{
    Public Void displayB(){
    System.out.println("Inside display B");
    }
}
class C extends B{
    public Void displayB(){
```

```
System.out.println("Inside display C");
}
}
Public class main{
    Public static Void main(String[] args){

        C obj=new();
        obj.displayA();
        obj.displayB();
          obj.displayC();
        }
```

Output:

Method from class A
Method from interface B
Method from interface C

Hybrid Inheritance:

```
class grandfather{
    Public Void show&(){
    System.out.println("He is grandfather");
}
class father extends Grandfather{
    Public Void showF(){
    System.out.println("He is father");
}
class Son extends father{
    public Void showS(){
```

```
System.out.println("He is son");
}
}
public class daughter extends father{
    Public Void showD(){
    System.out.println("she is daughter");
}
public static Void main(String args[]){
    Son obj= new Son();
    obj.showS();
    obj.showF();
    obj.showG();
    Daughter obj2= new daughter();
```

Output:

```
This animal eats food
the dog barks
the animal eats food
the cat meow.
```

Multiple Inheritance:

```
class A{
    Void method A(){
    System.out.println("Method from class A");
    }
}
```

```
Interface B{
    Void methodB();
}
Interface C{
    Void methodC();
}
Class D extends Implements B,C{
    Public Void methodB(){
System.out.println("Method from InterfaceB");
    }
    Public Void ~~Interface~~ methodC(){
System.out.println("She is daughter");
    }
Public Static Void main(String args[]){
    Son obj = new Son();
    obj.showS();
    obj.showF();
    obj.showC();
    Daughter obj2 = new daughter.
```