ASSIGNMENT

CSE 310 JAVA

Section: - K18MS

Name: - Vishal Kumar

Roll no: - B 58

Regis no: - 11803188

Problem No. Assigned: - 07

Submitted to: - Sudha Shankar Prasad (21826)

Professor

Department of Computer Science

Lovely Professional University

Jalandhar, Punjab

7. **Write a complete chapter on Inheritance in java which includes Definition of inheritance, importance, uses and its types. For each type you have to write a program which completely explains that type. Use inheritance with classes and interfaces.**

Ans:-

In programming language we can say that Inheritance is a process or mechanism in which one class acquires the property of another class. In simple words we can say that inherited means which we receive from our parents or other family members.
Inheritance is an important part of object oriented Programming language. Inheritance in java means that one object acquires all the properties and behaviour of a parent object or it is a process in which one class inherit all features of parent class or any another class.
Inheritance is also known as child-parent class relationship.

Importance of inheritance

The importances of inheritance in java are as follows:-

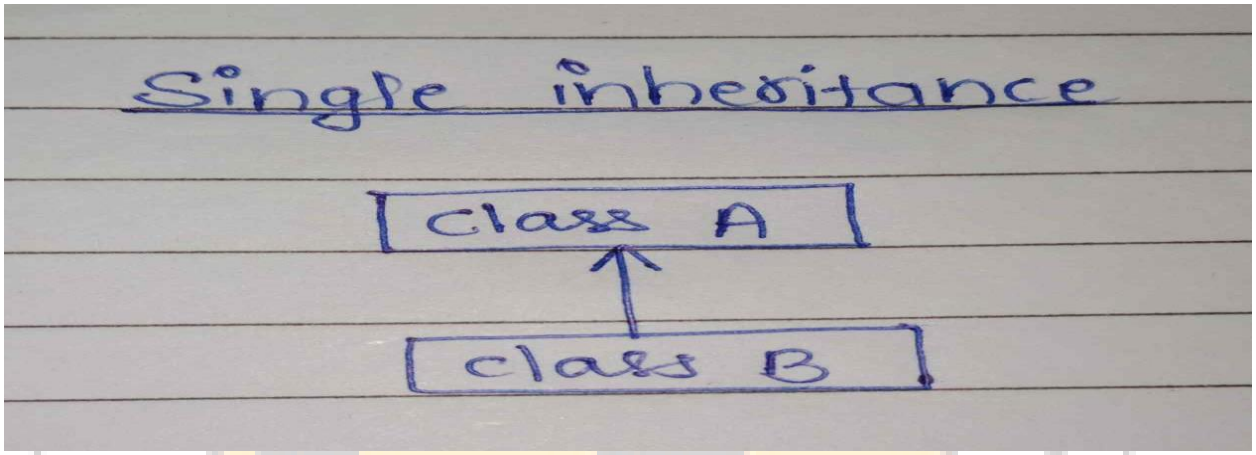
1. **Code Reusability**
   Inheritance helps in the reducing the rewriting of code. Code can be reused by other classes and the child class only has to write its own code or unique properties.
   This reduces the time consumption and complexity of the code.

2. **For method overriding**
   A method that is already present in the parent class is declared by the child class, and then it is known as method overriding.
   It is used to provides a particular implementation of a method that has already provided by its corresponding superclass.
   The concepts of method overriding also extends to runtime polymorphism.

Types of inheritance:

1. **Single Inheritance**

Single inheritance is a type of inheritance in which subclass inherits the features or property of one subclass.

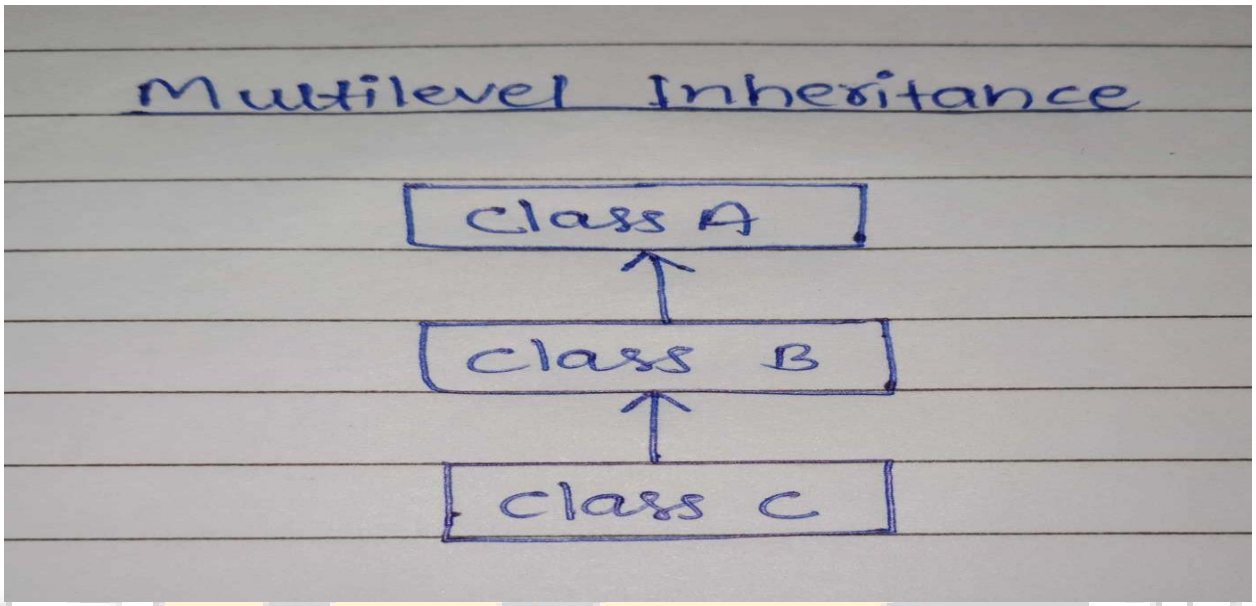


Programs of single inheritance

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

2. Multilevel Inheritance

Multilevel inheritance is a type of inheritance in which a derived class inherits a base class and a derived class that also acts as a base class.



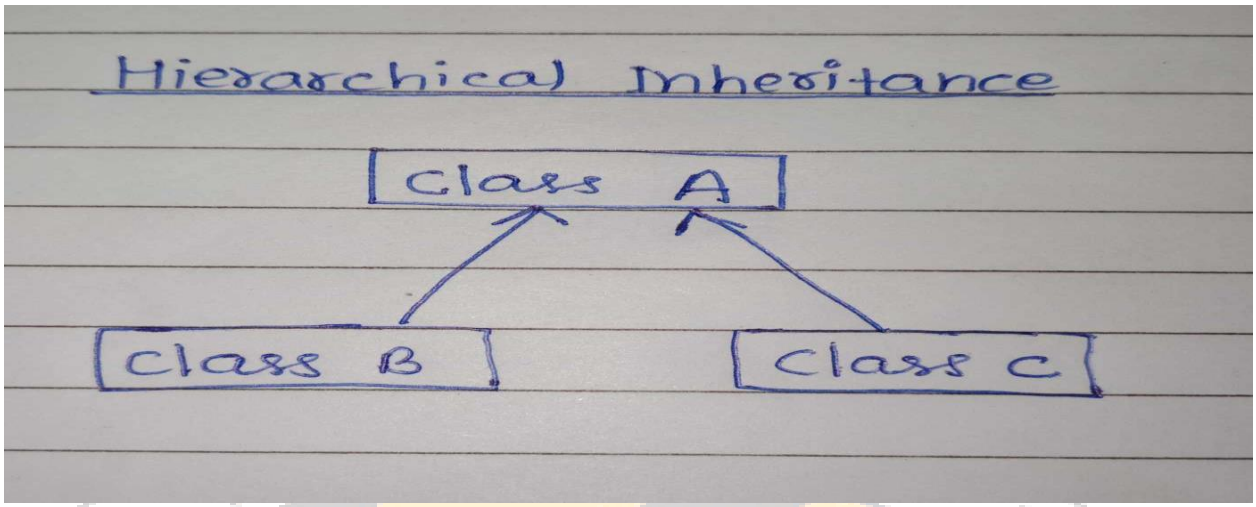
Programs of multilevel inheritance

```
class Car{
  public Car()
  {
        System.out.println("Class Car");
  }
  public void vehicleType()
  {
        System.out.println("Vehicle Type: Car");
  }
}
class Maruti extends Car{
  public Maruti()
  {
        System.out.println("Class Maruti");
  }
  public void brand()
  {
        System.out.println("Brand: Maruti");
```

```
    }
    public void speed()
    {
        System.out.println("Max: 90Kmph");
    }
}
public class Maruti800 extends Maruti{

    public Maruti800()
    {
        System.out.println("Maruti Model: 800");
    }
    public void speed()
    {
        System.out.println("Max: 80Kmph");
    }
    public static void main(String args [])
    {
        Maruti800 obj=new Maruti800();
        obj.vehicleType();
        obj.brand();
        obj.speed();
    }
}
```

3. Hierarchical Inheritance

Hierarchical inheritance is a type of inheritance in which one class acts as a base class for more than one sub class.



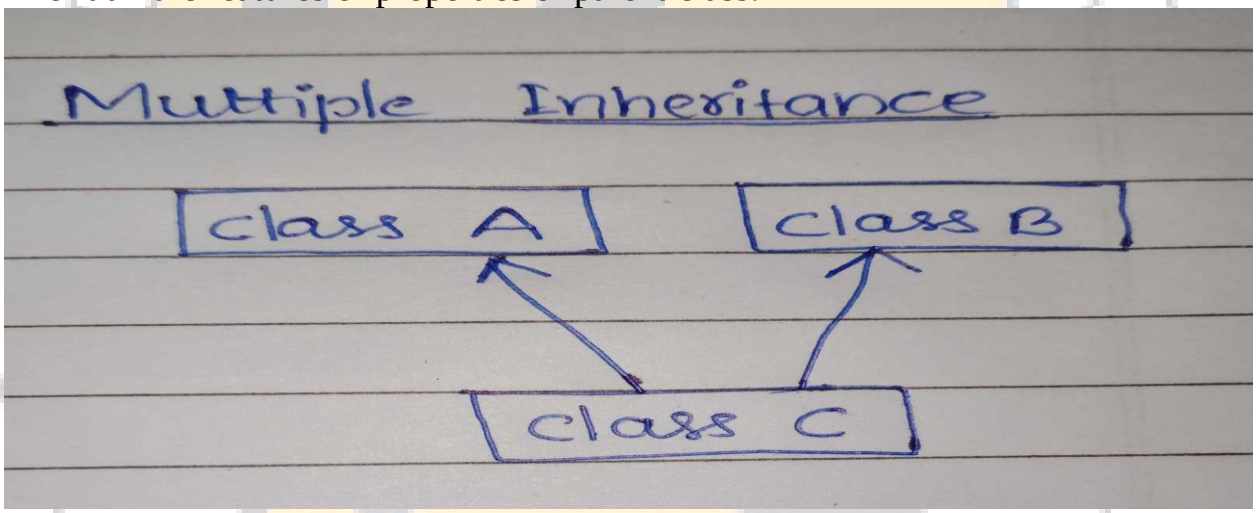
Programs of Hierarchical inheritance

```
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 TestInheritanceHierarchial{
```

```
    public static void main(String args[]){
    Cat c=new Cat();
    c.meow();
    c.eat();
    //c.bark(); //C.T.Error
    }
}
```

4. Multiple Inheritance

Multiple Inheritance do not support java with class.
It supports through interface.
It is type of inheritance in which one class have more than one superclass that inherit all the features or properties of parent class.



Programs of multiple inheritance

```
//Multiple Inheritance

class Animal
{
    void eat()
    {
            System.out.println("eating...");
    }
```

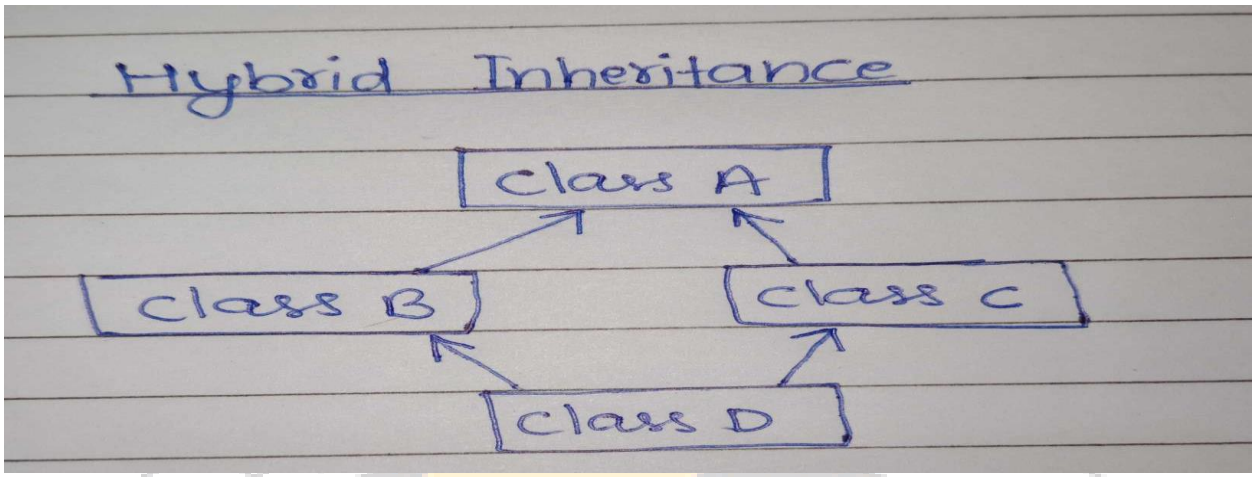
```
}
class Dog extends Animal
{
    void bark()
    {
        System.out.println("barking...");
    }
}
class BabyDog extends Dog
{
    void weep()
    {
        System.out.println("weeping...");
    }
}

class TestInheritanceMultiple
{
    public static void main(String args[]){
    BabyDog d=new BabyDog();
    d.weep();
    d.bark();
    d.eat();
    }
}
```

5. Hybrid Inheritance

It supports java through interfaces.

It is the combination of all inheritance like single, multiple, multilevel, hierarchical inheritance.

Java does not support the multiple inheritance with class. So we can says that hybrid inheritance will not support by java with class. It will supports through interfaces.



Programs of hybrid inheritance

```
interface A {
    public void planA();
}
interface B extends A {
    public void planB();
}
interface C extends A {
    public void planC();
}
class HybridInheritance implements B, C {
    public void planA() {
        System.out.println("Calling planA");

    }
    public void planB() {
        System.out.println("Calling planB");
    }
    public void planC() {
        System.out.println("Calling planC");
    }
    public static void main (String args[]) {
        HybridInheritance obj = new HybridInheritance();
        obj. planA();
        obj. planB();
        obj. planC();
    }
```

```
    }
```

Write a program to create a class Account (having attributes account_nos and balance) which inherits from Bank class (which contains two methods deposit and withdraw). Override the deposit method for cash and cheque. Create a class App which provides you a menu for deposit, withdraw, Account information and Exit. Perform all the above operations on the Account Number entered by the user.

```
import java.util.Scanner;

class Bank
{
    private String account_no;
    private String name;
    private long balance;

    Scanner sc=new Scanner(System.in);
    void openAccount()
    {
        System.out.print("Enter Account No:");
        account_no=sc.next();
        System.out.print("Enter Name:");
        name=sc.next();
        System.out.print("Enter Balance:");
        balance=sc.nextLong();
    }

    void showAccount()
    {
        System.out.println("Customer Account_No is:"+account_no+"\nCustomer Name is:"+name+"\nCustomer Bank Balance is:"+balance);
    }
```

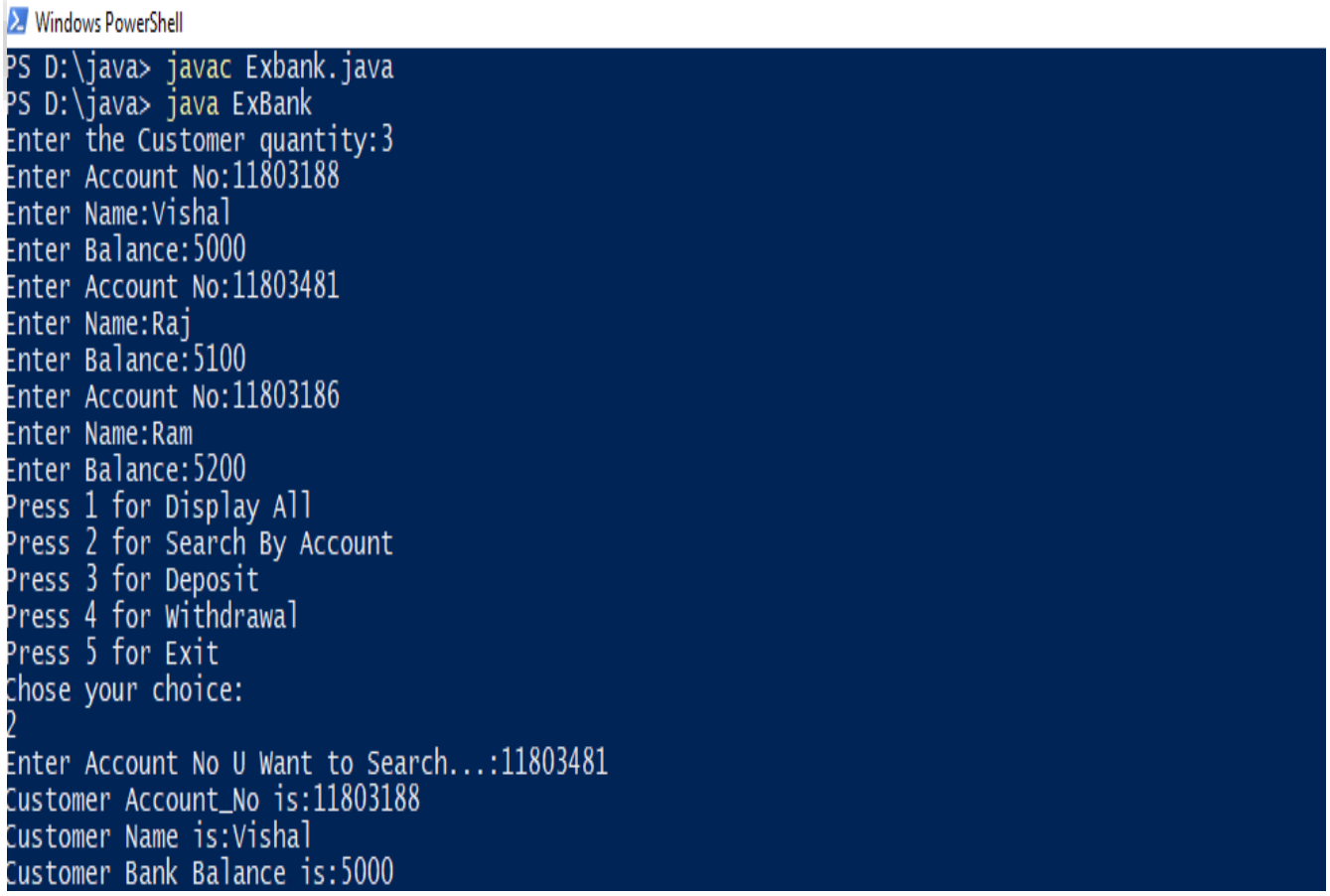
```
    void deposit()
    {
        long amount;
        System.out.println("Enter your Diposit amount:");
        amount=sc.nextLong();
        balance=balance+amount;
    }

    void withdrawal()
    {
        long amount;
        System.out.println("Enter your withdrawal amount:");
        amount=sc.nextLong();
        if(balance>=amount)
        {
            balance=balance-amount;
        }
        else
        {
            System.out.println("Less Balance transaction Failed.....");
        }
    }

    boolean search(String account_no)
    {
        if(account_no.equals(account_no))
        {
            showAccount();
            return(true);
        }
        return(false);
    }
}

class ExBank
{
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter the Customer quantity:");
```

```
int n=sc.nextInt();
Bank  C[]=new Bank[n];
for(int i=0;i<C.length;i++)
{
	C[i]=new Bank();
	C[i].openAccount();
}

int ch;
do
{
	System.out.println("Press 1 for Display All\nPress 2 for Search By Account\nPress 3 for Deposit\nPress 4 for Withdrawal\nPress 5 for Exit");
	System.out.println("Chose your choice:");
	ch=sc.nextInt();
	switch(ch)
	{
		case 1:
			for(int i=0;i<C.length;i++)
			{
				C[i].showAccount();
			}
			break;

		case 2:
			System.out.print("Enter Account No U Want to Search...:");
			String account_no=sc.next();
			boolean found=false;
			for(int i=0;i<C.length;i++)
			{
				found=C[i].search(account_no);
				if(found)
				{
					break;
				}
			}
			if(!found)
			{
```

```
                    System.out.println("Search Failed..Account Not Exist..");
                }
                break;

            case 3:
                System.out.print("Enter Account No:");
                account_no=sc.next();
                found=false;
                for(int i=0;i<C.length;i++)
                {
                    found=C[i].search(account_no);
                    if(found)
                    {
                        C[i].deposit();
                        break;
                    }
                }
                if(!found)
                {
                    System.out.println("Search Failed..Account Not Exist..");
                }
                break;

            case 4:
                System.out.print("Enter Account No:");
                account_no=sc.next();
                found=false;
                for(int i=0;i<C.length;i++)
                {
                    found=C[i].search(account_no);
                    if(found)
                    {
                        C[i].withdrawal();
                        break;
                    }
                }
                if(!found)
                {
```

```
                                    System.out.println("Search Failed..Account Not Exist..");
                                }
                                break;

                            case 5:
                                System.out.println("Thanks for visiting.....Have a Nice Day");
                                break;
                        }
                    }
                    while(ch!=5);
                }
            }
```

Outputs:-



```
Press 1 for Display All
Press 2 for Search By Account
Press 3 for Deposit
Press 4 for Withdrawal
Press 5 for Exit
Chose your choice:
1
Customer Account_No is:11803188
Customer Name is:Vishal
Customer Bank Balance is:5000
Customer Account_No is:11803481
Customer Name is:Raj
Customer Bank Balance is:5100
Customer Account_No is:11803186
Customer Name is:Ram
Customer Bank Balance is:5200
Press 1 for Display All
Press 2 for Search By Account
Press 3 for Deposit
Press 4 for Withdrawal
Press 5 for Exit
Chose your choice:
```

```
Press 1 for Display All
Press 2 for Search By Account
Press 3 for Deposit
Press 4 for Withdrawal
Press 5 for Exit
Chose your choice:
3
Enter Account No:11803188
Customer Account_No is:11803188
Customer Name is:Vishal
Customer Bank Balance is:5000
Enter your Diposit amount:
5000
Press 1 for Display All
Press 2 for Search By Account
Press 3 for Deposit
Press 4 for Withdrawal
Press 5 for Exit
Chose your choice:
2
Enter Account No U Want to Search...:11803188
Customer Account_No is:11803188
Customer Name is:Vishal
Customer Bank Balance is:10000
Press 1 for Display All
Press 2 for Search By Account
Press 3 for Deposit
Press 4 for Withdrawal
Press 5 for Exit
Chose your choice:
```

```
Press 1 for Display All
Press 2 for Search By Account
Press 3 for Deposit
Press 4 for Withdrawal
Press 5 for Exit
Chose your choice:
4
Enter Account No:11803481
Customer Account_No is:11803188
Customer Name is:Vishal
Customer Bank Balance is:10000
Enter your withdrawal amount:
200
Press 1 for Display All
Press 2 for Search By Account
Press 3 for Deposit
Press 4 for Withdrawal
Press 5 for Exit
Chose your choice:
2
Enter Account No U Want to Search...:11803481
Customer Account_No is:11803188
Customer Name is:Vishal
Customer Bank Balance is:9800
Press 1 for Display All
Press 2 for Search By Account
Press 3 for Deposit
Press 4 for Withdrawal
Press 5 for Exit
Chose your choice:
```

Thank you Sir