Write a simple Java programme to complete the following requirement.

There is a small bank which has the two account types. The details of those two account type as followed.

Name	Isuru
Deposit Method	The bank will be adding 5% bonus for each deposit amount.
Withdraw Method	The band will be charging Rs 5.00 for each withdrawal.

Name	Nirogya
Deposit Method	The bank will be adding 10% bonus for each deposit amount.
Withdraw Method	The band will be charging Rs 5.00for each withdrawal.

You have to complete the following task.

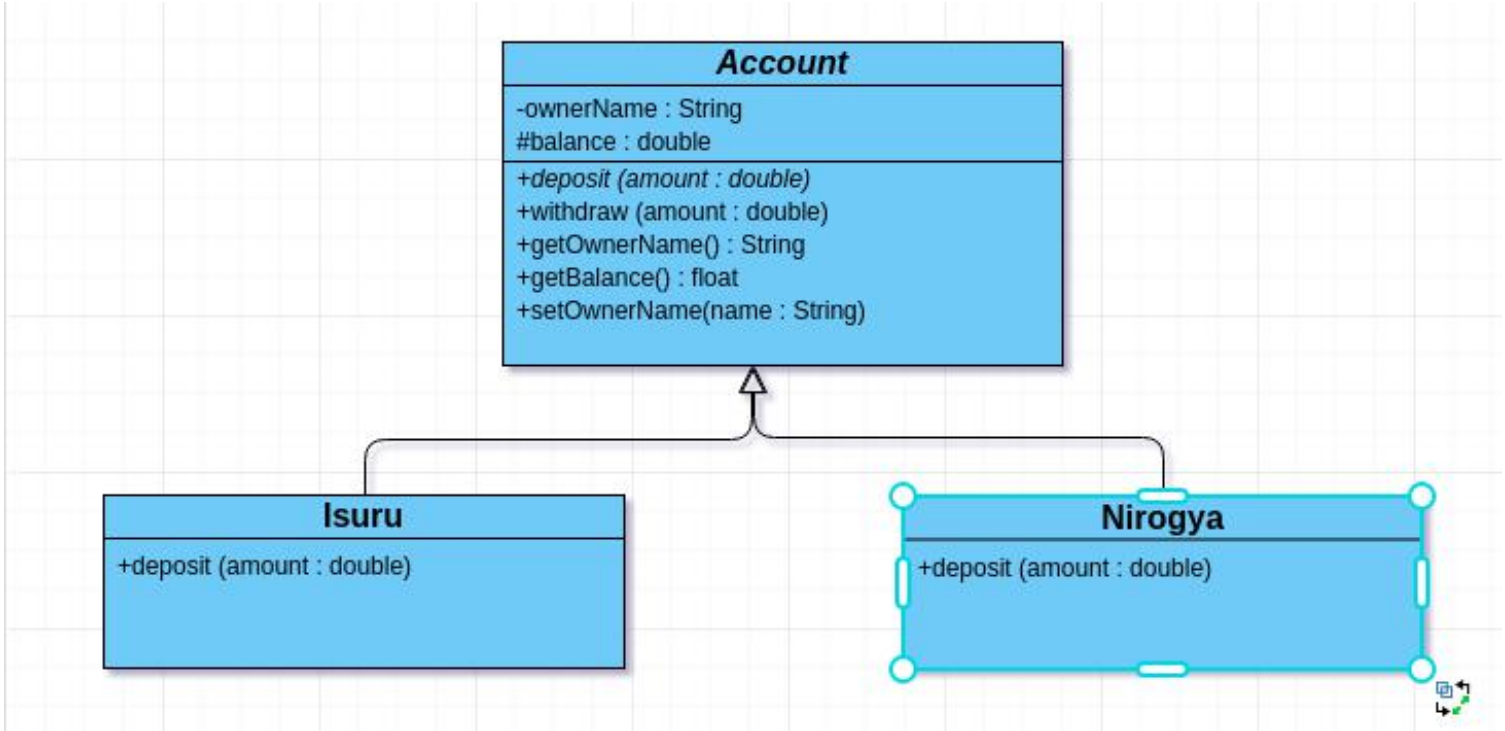
- Draw a class diagram for the above scenario
- You have to create proper code structure using the proper OOP concept.
- Complete the implementation of the withdraw and deposit methods.
- Finally you have to create objects of any account type and call the deposit and withdraw methods.
- Please note that there should be proper loggings for every class.

Note: please contact your customer if requirements are not clear

Answers for the OOP Assessment

- **Account** is a abstract class that having these two methods.And that is the parent class.It is having a abstract method called "**deposit**" and implemented method called "**withdraw** ".Because Class Isuru and **Nirogya** are having the same "**withdraw**" methods but different "**deposit**" methods. **Account** class is a **abstract class** since it has at least one unimplemented(abstract) method.
- class **Isuru** and **Nirogya** extend **Account** class.

Class Diagram



Note:- In here **Account** is a **abstract** class , so it is in the ***italic*** text format. In the account class there is a abstract method called "deposit" and it is also in ***italic*** text format. It has a concrete method called "withdraw". In class "Isuru" and "Nirogya" are concrete classes and they have implemented the "deposit" method in each of this classes.

- Two variables in **Account** class. They are **ownerName**(String) and **balance**(double). I made **ownerName** as private and **balance** as protected since in extended classes we need to access **balance** variable when we are working with **deposit** and **withdraw** methods. Then it can only be access to the extended classes and not to the outer classes.
- I have provide **getOwnerName()** and **getBalance()** methods to get the **ownerName** and **balance** variables for outer classes. And I have provided **setOwnerName(String name)** method to set the name variable from outer classes (Since at the begging also we need to provide a name. So outer classes which are not extended from **Account** class also can set a value for the name variable).

Code

```
abstract class Account{
    private String ownerName;
    protected double balance;
    // initializing the constructor class
    Account(){
        this.ownerName = "jayakodi";
        this.balance = 0;
    }
    abstract void deposit(double amount);
    // implementing the withdraw method
    public void withdraw(double amount){
    balance -= (amount + 5);
    }
```

```
    public String getOwnerName(){
        return ownerName;
    }
    public double getBalance(){
        return balance;
    }
    public void setOwnerName(String name){
        this.ownerName = name;
    }
}
class Isuru extends Account{
// implementation of the deposit method
    public void deposit(double amount){
    balance += amount + amount*0.05;
    }
}
class Nirogya extends Account{
// implementation of the deposit method
    public void deposit(double amount){
    balance += amount + amount*0.1;
    }
}
public class Main{
    public static void main(String []args){
        double amount = 1000;
        String name = "prasadi";
// creating a Isuru type object
        Isuru isuru = new Isuru();
        isuru.setOwnerName(name);
```

```
// calling the both deposit and withdraw methods
            isuru.deposit(amount);
             isuru.withdraw(amount);
             System.out.println(isuru.getOwnerName());
             System.out.println(isuru.getBalance());
        }
}

// Output
```