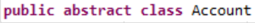
Credit Name: CSE3130 Object Oriented Programming 2

Assignment Name: Account

How has your program changed from planning to coding to now? Please explain?

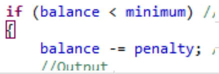
After reading the instructions in the textbook I knew I recognized the assignment. Because of this, I found my code from chapter 10 for LocalBank and decided to use it as the mentioned client code, modifying it to use inheritance.

I started by making Account abstract



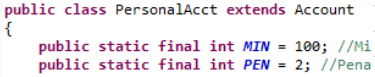
```
public abstract class Account
```

I then added the minimum and penalty variables, and implemented them into withdraw()



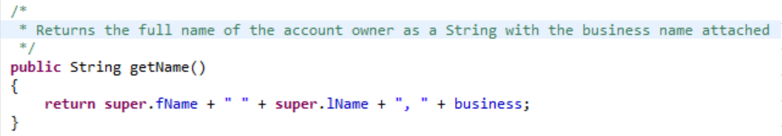
```
 if (balance < minimum) /
 {
     balance -= penalty; /
     //Output
```

Then I created the subclasses which have their own constant minimum and penalty variables which they automatically pass into the superconstructor



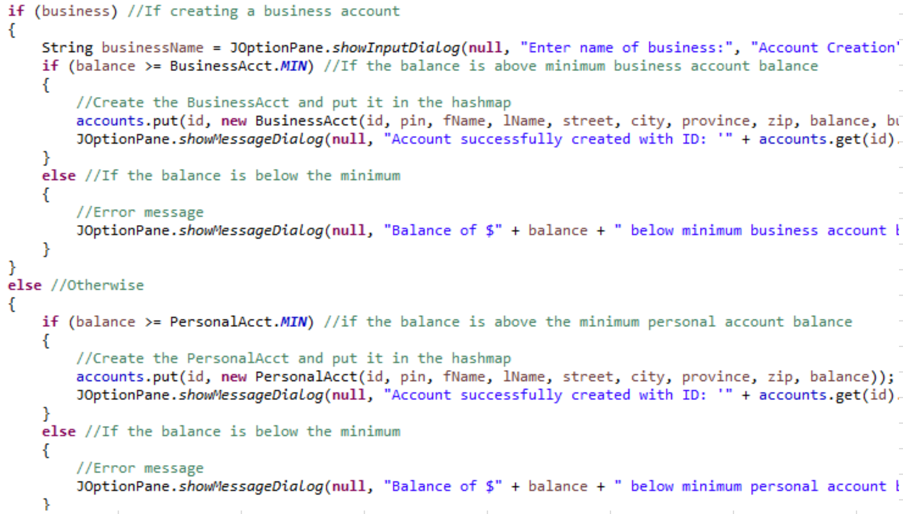
```
public class PersonalAcct extends Account
{
    public static final int MIN = 100; //Mi
    public static final int PEN = 2; //Pena
```

Business accounts also have a variable for business name which is obtained from the constructor and an overridden getName() that returns it with the business name on the end



```
/*
 * Returns the full name of the account owner as a String with the business name attached
 */
public String getName()
{
    return super.fName + " " + super.lName + ", " + business;
}
```

Bank was modified to create either a personal or business account as desired instead of an abstract account, and to ensure that the minimum balance is present



```
if (business) //If creating a business account
{
    String businessName = JOptionPane.showInputDialog(null, "Enter name of business:", "Account Creation'
    if (balance >= BusinessAcct.MIN) //If the balance is above minimum business account balance
    {
        //Create the BusinessAcct and put it in the hashmap
        accounts.put(id, new BusinessAcct(id, pin, fName, lName, street, city, province, zip, balance, bu
        JOptionPane.showMessageDialog(null, "Account successfully created with ID: '" + accounts.get(id).
    }
    else //If the balance is below the minimum
    {
        //Error message
        JOptionPane.showMessageDialog(null, "Balance of $" + balance + " below minimum business account b
    }
}
else //Otherwise
{
    if (balance >= PersonalAcct.MIN) //if the balance is above the minimum personal account balance
    {
        //Create the PersonalAcct and put it in the hashmap
        accounts.put(id, new PersonalAcct(id, pin, fName, lName, street, city, province, zip, balance));
        JOptionPane.showMessageDialog(null, "Account successfully created with ID: '" + accounts.get(id).
    }
    else //If the balance is below the minimum
    {
        //Error message
        JOptionPane.showMessageDialog(null, "Balance of $" + balance + " below minimum personal account b
    }
```

LocalBank was modified in a couple places to gather modified input and show modified output to match the new account types

When I was commenting the code I also realized that this older project does not use correct method comments from the code standards, so I went through and updated all of them.

Overall it was far more interesting than the other projects, as there was actually something to be done instead of creating an OOP system for the sake of it. Revisiting some old code was interesting as well.

The subclasses at first were quite boring and felt unnecessary, but I think that adding the business name gives it enough variation that it doesn't feel pointless to have them. The framework would also allow more changes in the future which is always helpful.