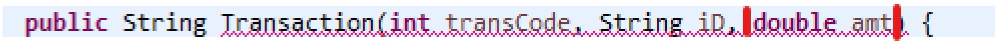
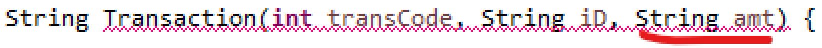
Credit Name: CSE3010 - Computer Science 3

Assignment Name: LocalBankGUI

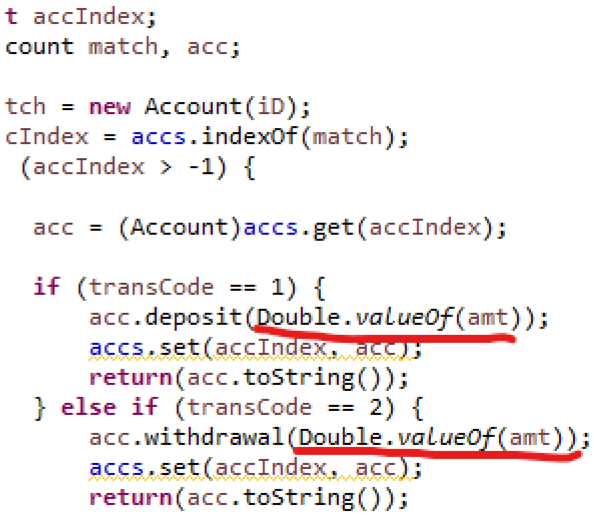
**Error** - “Must return type String”, this error would appear when I added the “double amt” into the parameters, the other two would be fine until I added the third



```
 public String Transaction(int transCode, String iD, double amt) {
```

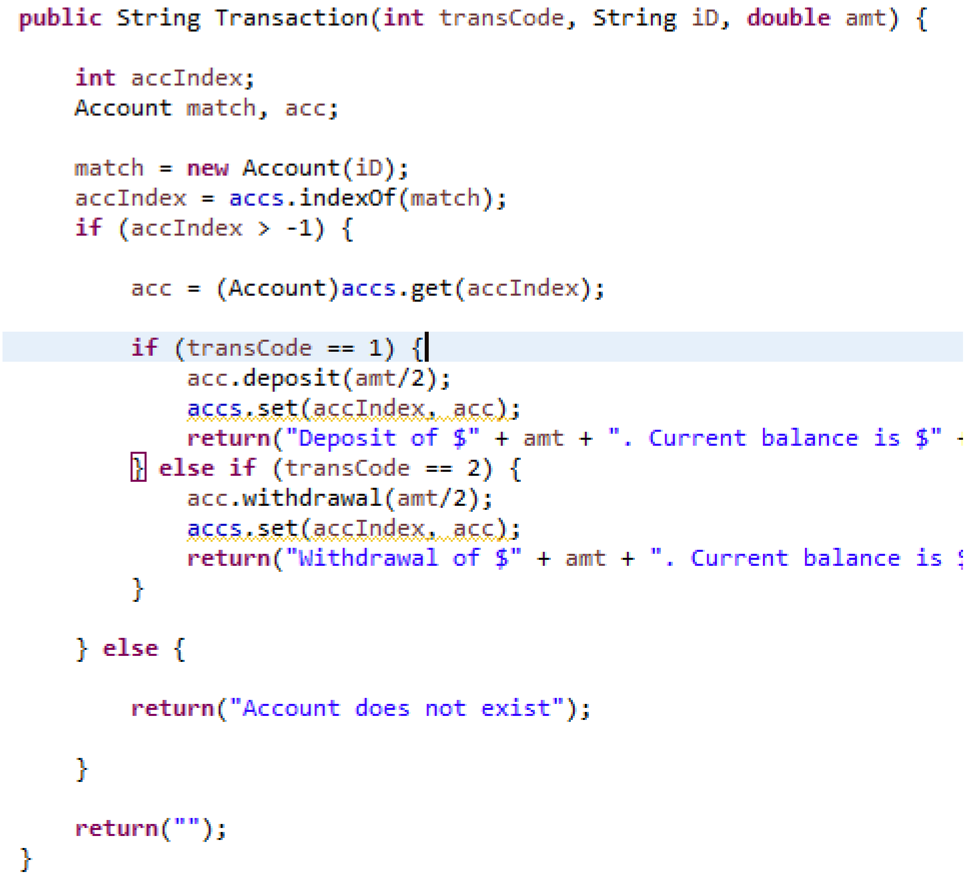


```
 String Transaction(int transCode, String iD, String amt) {
```



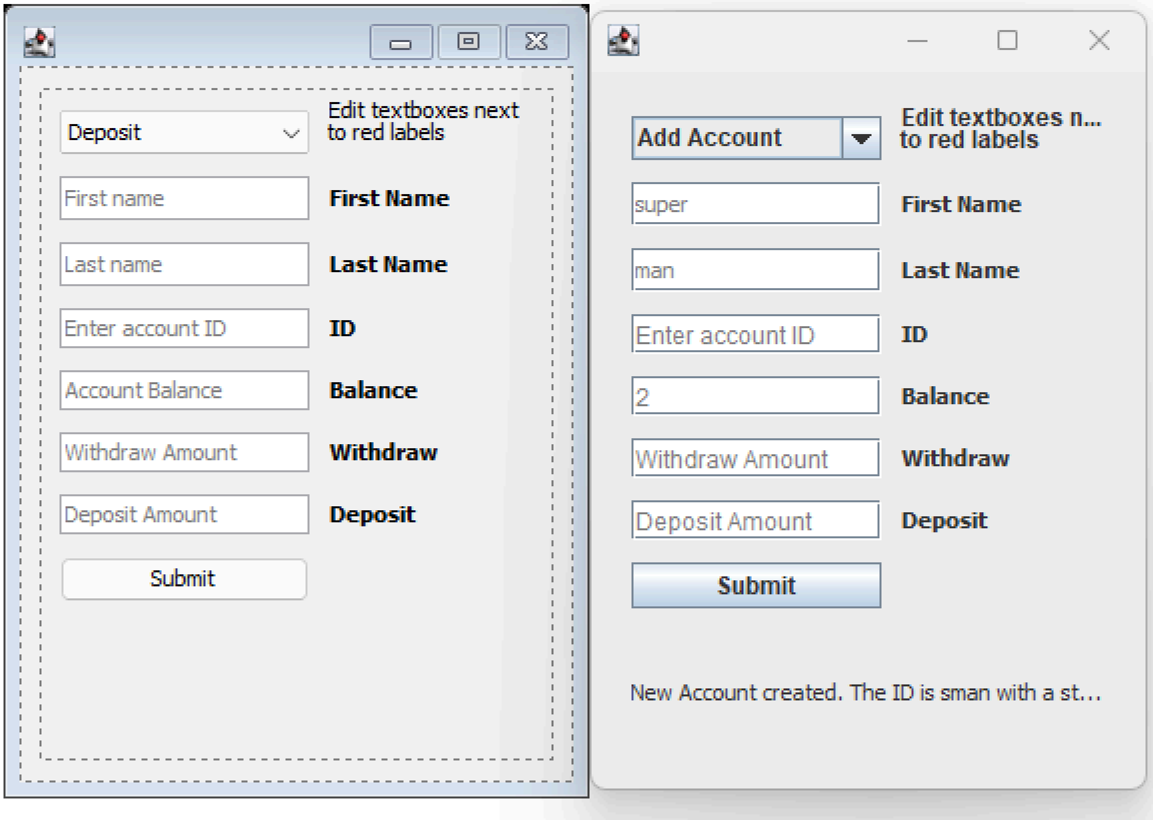
```
t accIndex;
count match, acc;

tch = new Account(iD);
cIndex = accs.indexOf(match);
 (accIndex > -1) {

  acc = (Account)accs.get(accIndex);

  if (transCode == 1) {
      acc.deposit(Double.valueOf(amt));
      accs.set(accIndex, acc);
      return(acc.toString());
  } else if (transCode == 2) {
      acc.withdrawal(Double.valueOf(amt));
      accs.set(accIndex, acc);
      return(acc.toString());
```

**Solution** - added a return(“”) that returned an empty string.



```
public String Transaction(int transCode, String iD, double amt) {

    int accIndex;
    Account match, acc;

    match = new Account(iD);
    accIndex = accs.indexOf(match);
    if (accIndex > -1) {

        acc = (Account)accs.get(accIndex);

        if (transCode == 1) {
            acc.deposit(amt/2);
            accs.set(accIndex, acc);
            return("Deposit of $" + amt + ". Current balance is $" +
        } else if (transCode == 2) {
            acc.withdrawal(amt/2);
            accs.set(accIndex, acc);
            return("Withdrawal of $" + amt + ". Current balance is $
        }

    } else {

        return("Account does not exist");

    }

    return("");
}
```

**Error** - Words would cut off the GUI and weren't aesthetically pleasing. This was because I used a jLabel instead of a JTextArea.



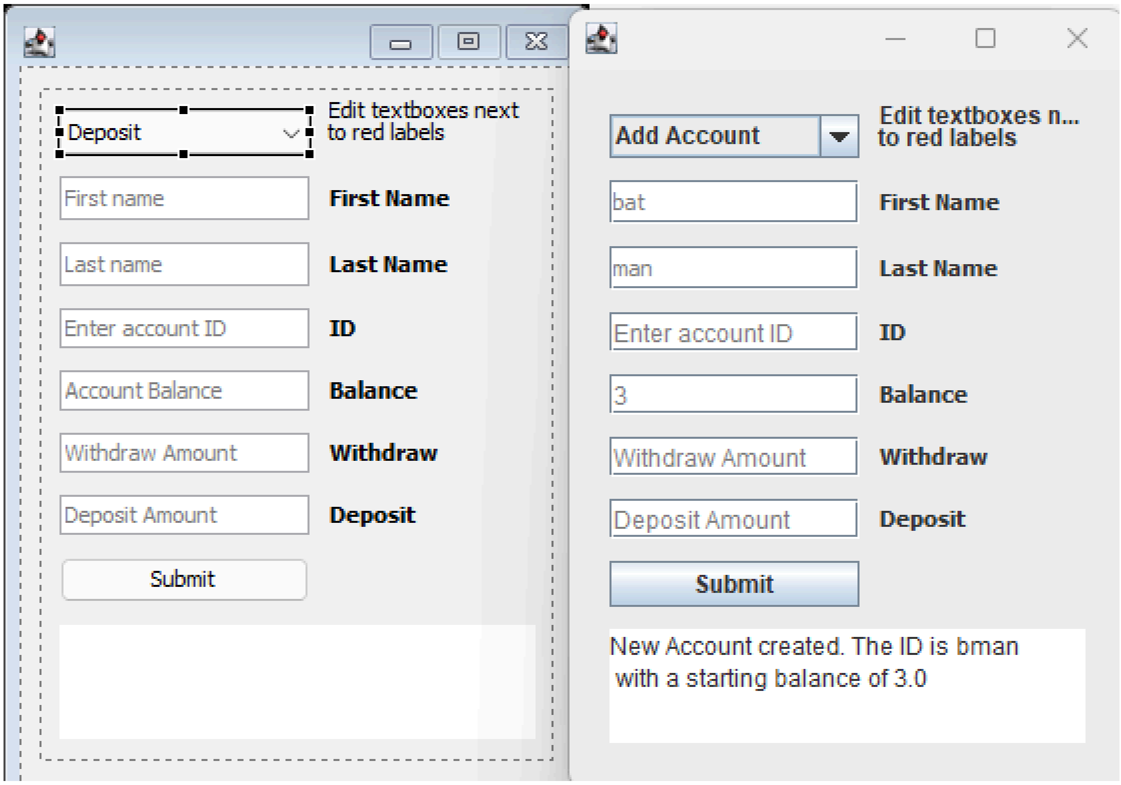
First Name: super

Last Name: man

Balance: 2

New Account created. The ID is sman with a st…

**Solution** - Replaced with a JTextArea



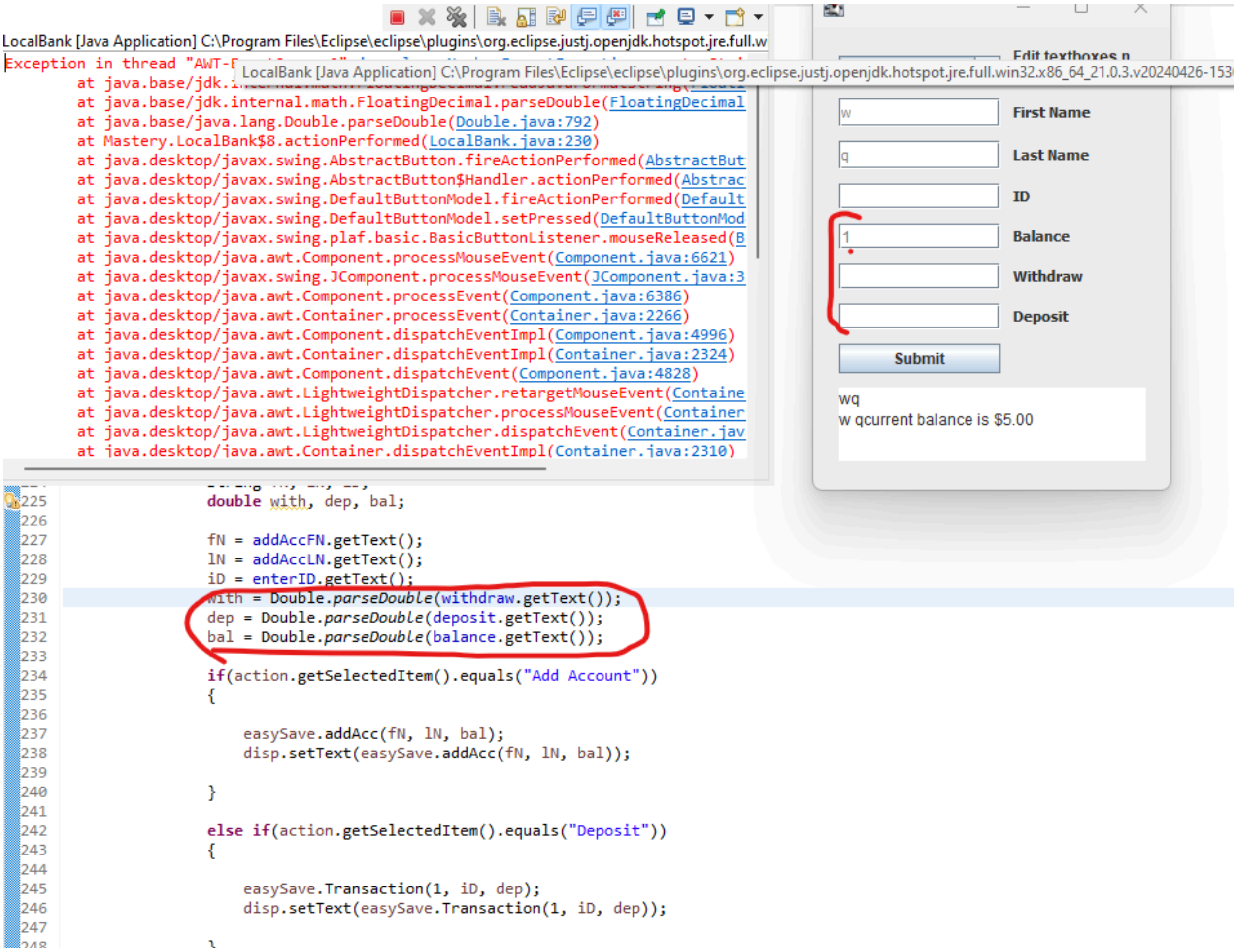
First Name: bat

Last Name: man

Balance: 3

```
New Account created. The ID is bman
 with a starting balance of 3.0
```

**Error** - Tried to put variables at the start of the main code under action performed. This was for organization. This wouldn't work because if the user pressed submit with an empty balance, withdraw, or deposit field then this error would appear since the string couldn't be parsed into a double.



```
LocalBank [Java Application] C:\Program Files\Eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.w
Exception in thread "AWT-E
        at java.base/jdk.i
        at java.base/jdk.internal.math.FloatingDecimal.parseDouble(FloatingDecimal
        at java.base/java.lang.Double.parseDouble(Double.java:792)
        at Mastery.LocalBank$8.actionPerformed(LocalBank.java:230)
        at java.desktop/javax.swing.AbstractButton.fireActionPerformed(AbstractBut
        at java.desktop/javax.swing.AbstractButton$Handler.actionPerformed(Abstrac
        at java.desktop/javax.swing.DefaultButtonModel.fireActionPerformed(Default
        at java.desktop/javax.swing.DefaultButtonModel.setPressed(DefaultButtonMod
        at java.desktop/javax.swing.plaf.basic.BasicButtonListener.mouseReleased(B
        at java.desktop/java.awt.Component.processMouseEvent(Component.java:6621)
        at java.desktop/javax.swing.JComponent.processMouseEvent(JComponent.java:3
        at java.desktop/java.awt.Component.processEvent(Component.java:6386)
        at java.desktop/java.awt.Container.processEvent(Container.java:2266)
        at java.desktop/java.awt.Component.dispatchEventImpl(Component.java:4996)
        at java.desktop/java.awt.Container.dispatchEventImpl(Container.java:2324)
        at java.desktop/java.awt.Component.dispatchEvent(Component.java:4828)
        at java.desktop/java.awt.LightweightDispatcher.retargetMouseEvent(Containe
        at java.desktop/java.awt.LightweightDispatcher.processMouseEvent(Container
        at java.desktop/java.awt.LightweightDispatcher.dispatchEvent(Container.jav
        at java.desktop/java.awt.Container.dispatchEventImpl(Container.java:2310)
```

LocalBank [Java Application] C:\Program Files\Eclipse\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.3.v20240426-153

First Name: w

Last Name: q

ID:

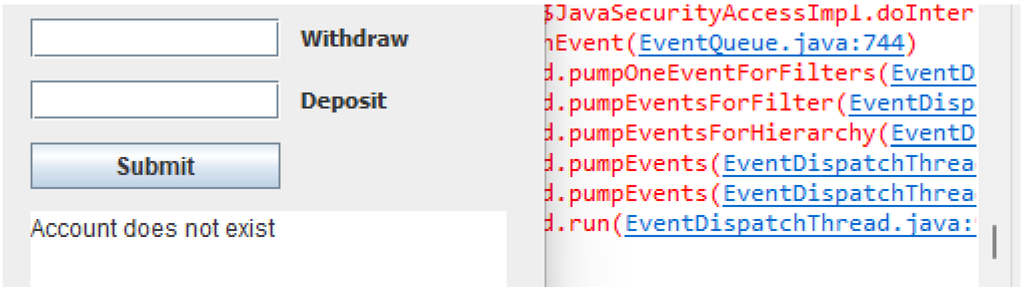
Balance: 1

Withdraw:

Deposit:

```
wq
w qcurrent balance is $5.00
```

```java
            double with, dep, bal;

            fN = addAccFN.getText();
            lN = addAccLN.getText();
            iD = enterID.getText();
            with = Double.parseDouble(withdraw.getText());
            dep = Double.parseDouble(deposit.getText());
            bal = Double.parseDouble(balance.getText());

            if(action.getSelectedItem().equals("Add Account"))
            {

                easySave.addAcc(fN, lN, bal);
                disp.setText(easySave.addAcc(fN, lN, bal));

            }

            else if(action.getSelectedItem().equals("Deposit"))
            {

                easySave.Transaction(1, iD, dep);
                disp.setText(easySave.Transaction(1, iD, dep));

            }
```



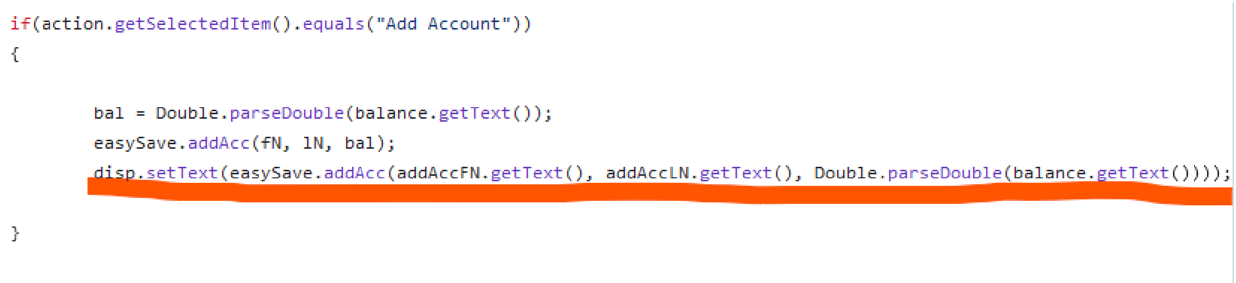
Withdraw:

Deposit:

```
Account does not exist
```

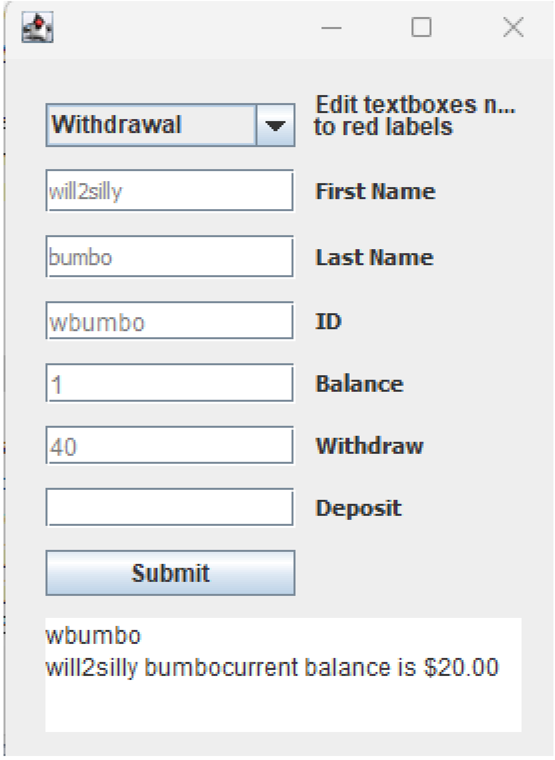
```
$JavaSecurityAccessImpl.doInter
nEvent(EventQueue.java:744)
d.pumpOneEventForFilters(EventD
d.pumpEventsForFilter(EventDisp
d.pumpEventsForHierarchy(EventD
d.pumpEvents(EventDispatchThrea
d.pumpEvents(EventDispatchThrea
d.run(EventDispatchThread.java:
```

**Solution** - I saved space and got rid of the variables and instead put them into the parameters of the method.



```
if(action.getSelectedItem().equals("Add Account"))
{

        bal = Double.parseDouble(balance.getText());
        easySave.addAcc(fN, lN, bal);
        disp.setText(easySave.addAcc(addAccFN.getText(), addAccLN.getText(), Double.parseDouble(balance.getText())));

}
```

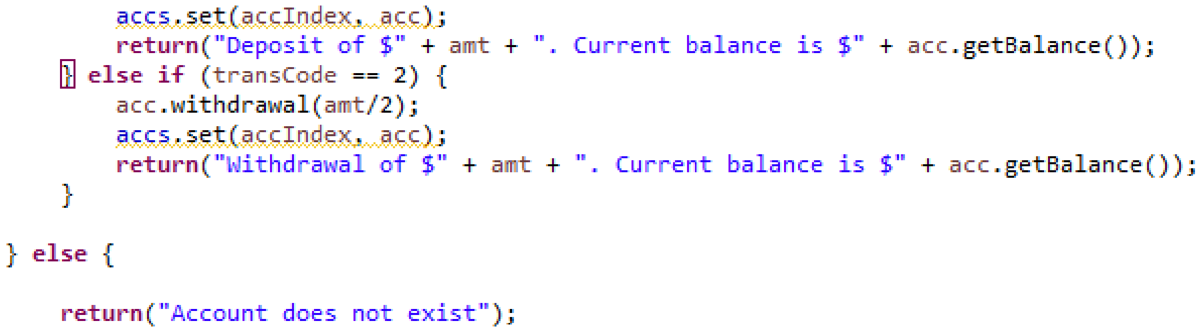
**Error** - No error message when user can't withdrawal



| | |
|---|---|
| Withdrawal ▾ | Edit textboxes n... to red labels |
| will2silly | First Name |
| bumbo | Last Name |
| wbumbo | ID |
| 1 | Balance |
| 40 | Withdraw |
| | Deposit |

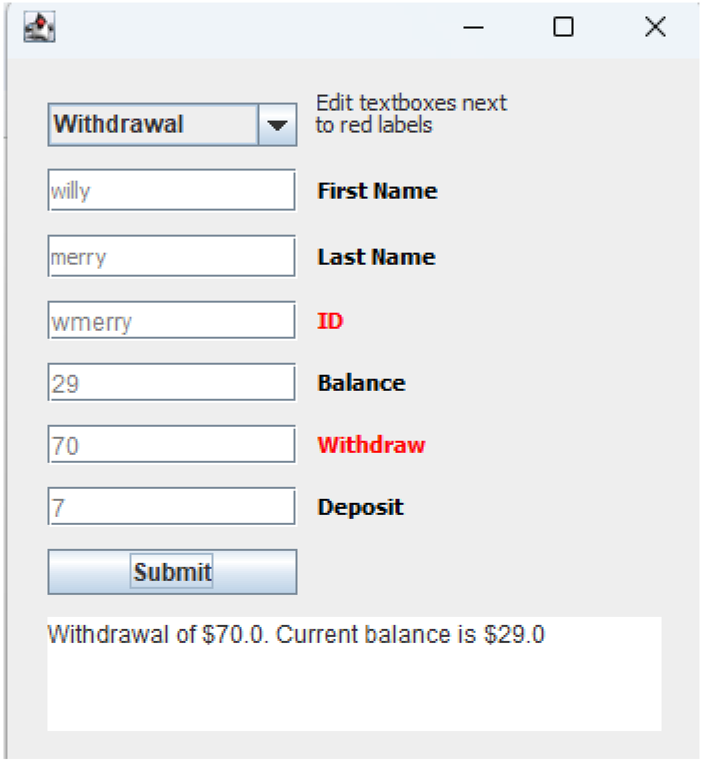
```
wbumbo
will2silly bumbocurrent balance is $20.00
```

**Attempt 1** - Edited code. Before, the code returned the ToString method now it just returned a string.



```
        accs.set(accIndex, acc);
        return("Deposit of $" + amt + ". Current balance is $" + acc.getBalance());
    } else if (transCode == 2) {
        acc.withdrawal(amt/2);
        accs.set(accIndex, acc);
        return("Withdrawal of $" + amt + ". Current balance is $" + acc.getBalance());
    }

} else {

    return("Account does not exist");
```

**Error** - Wouldn't display error message and instead it just wouldn't perform withdrawal and display as normal.



Withdrawal

Edit textboxes next to red labels

willy First Name

merry Last Name

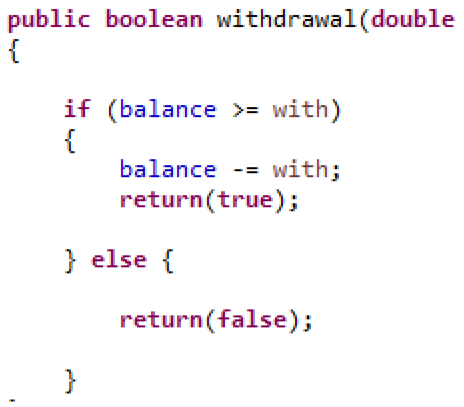
wmerry ID

29 Balance

70 Withdraw

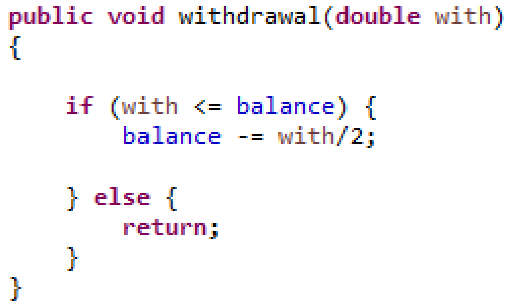
7 Deposit

Withdrawal of $70.0. Current balance is $29.0

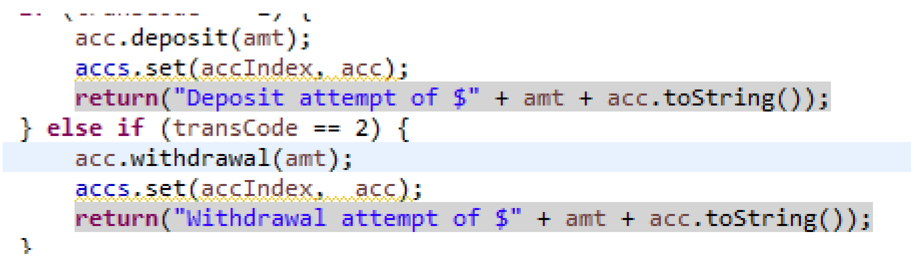


```
public boolean withdrawal(double
{

    if (balance >= with)
    {
        balance -= with;
        return(true);

    } else {

        return(false);

    }
```

**Solution** - Changed the wording on the withdrawal and turned the method to a void method..

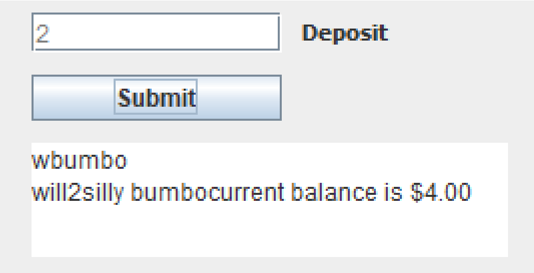


```
public void withdrawal(double with)
{

    if (with <= balance) {
        balance -= with/2;

    } else {
        return;
    }
}
```



```
    acc.deposit(amt);
    accs.set(accIndex, acc);
    return("Deposit attempt of $" + amt + acc.toString());
} else if (transCode == 2) {
    acc.withdrawal(amt);
    accs.set(accIndex,  acc);
    return("Withdrawal attempt of $" + amt + acc.toString());
```

**Error** - Withdraw and deposit would be 2 times that of the inputted number in corresponding JTextField.



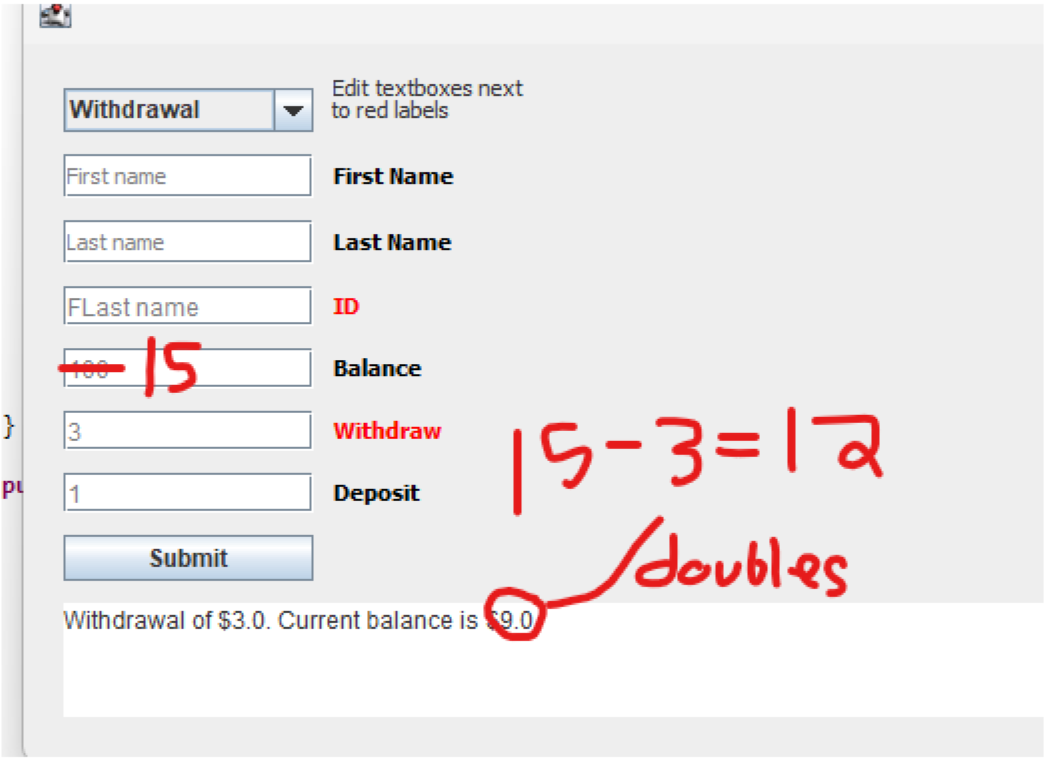
2 Deposit

```
wbumbo
will2silly bumbocurrent balance is $4.00
```

- Starting balance was 0 in this pic.

Let x = Number in Deposit JTextField.

dep = 2x<br>
dep/2 = 1.5x<br>
dep/3 = 1.3 repeating x<br>
dep/10 = 1.1x



Withdrawal

Edit textboxes next to red labels

First name First Name

Last name Last Name

FLast name ID

~~100~~ 15 Balance

3 Withdraw

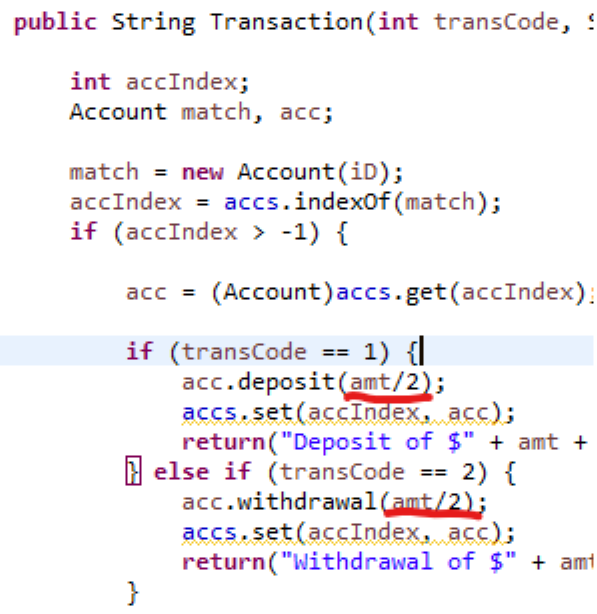
1 Deposit

```
Withdrawal of $3.0. Current balance is $9.0
```

15-3=12

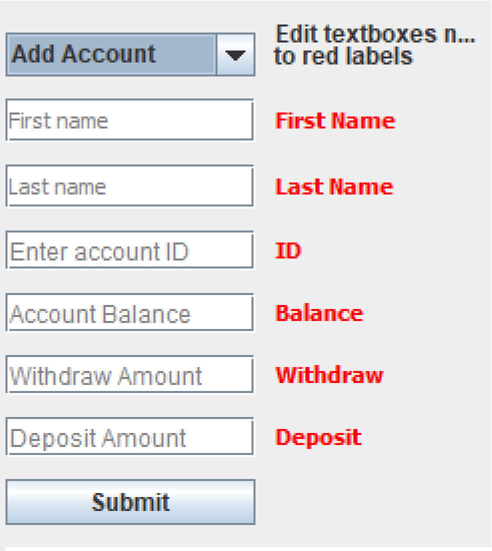
doubles

**Solution** - Had to edit the bank class and divide amt by 2 in the Transaction method.



```
public String Transaction(int transCode, S

    int accIndex;
    Account match, acc;

    match = new Account(iD);
    accIndex = accs.indexOf(match);
    if (accIndex > -1) {

        acc = (Account)accs.get(accIndex);

        if (transCode == 1) {
            acc.deposit(amt/2);
            accs.set(accIndex, acc);
            return("Deposit of $" + amt +
        } else if (transCode == 2) {
            acc.withdrawal(amt/2);
            accs.set(accIndex, acc);
            return("Withdrawal of $" + amt
        }
```

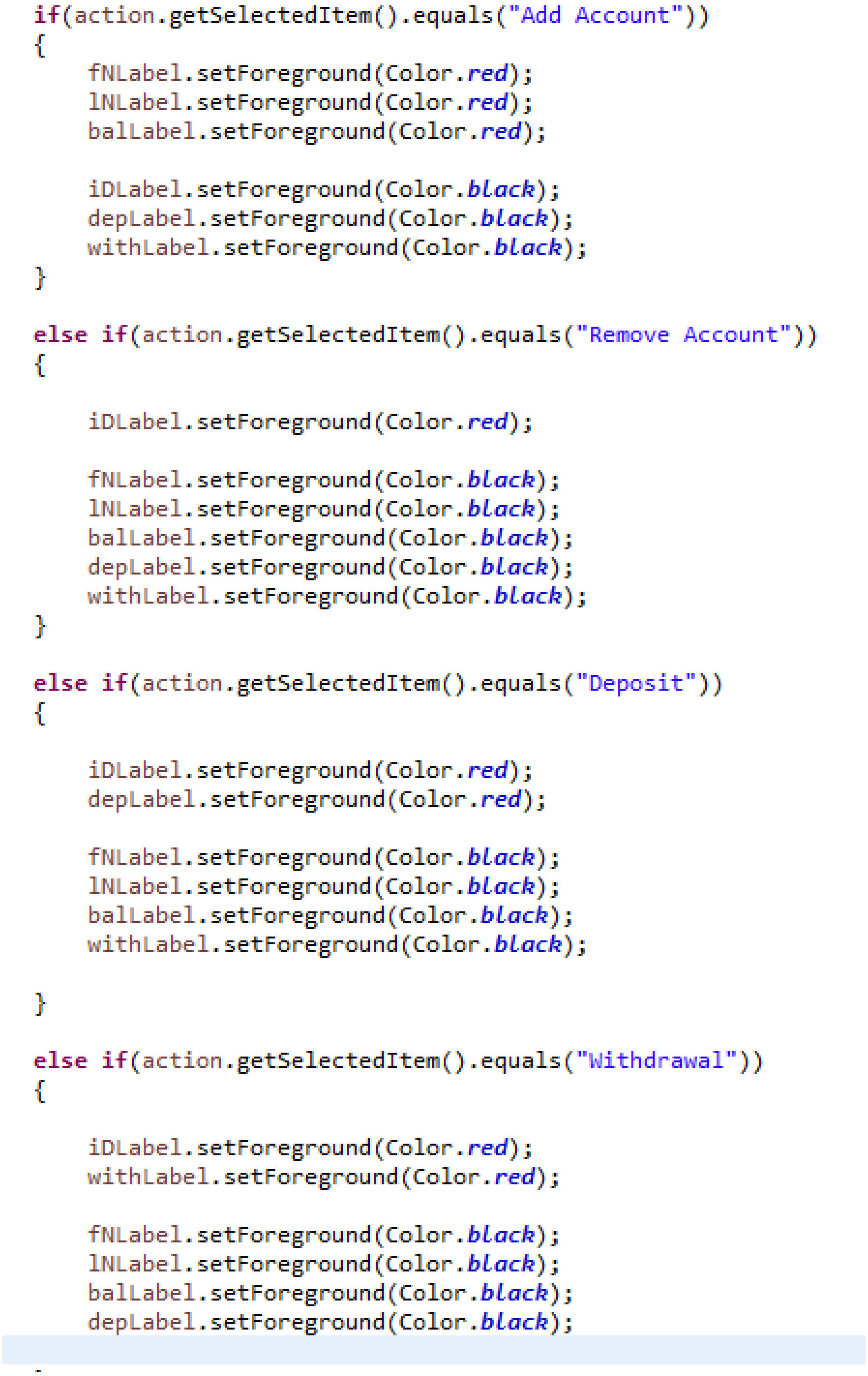
**Error** - Labels would turn red and stay red.



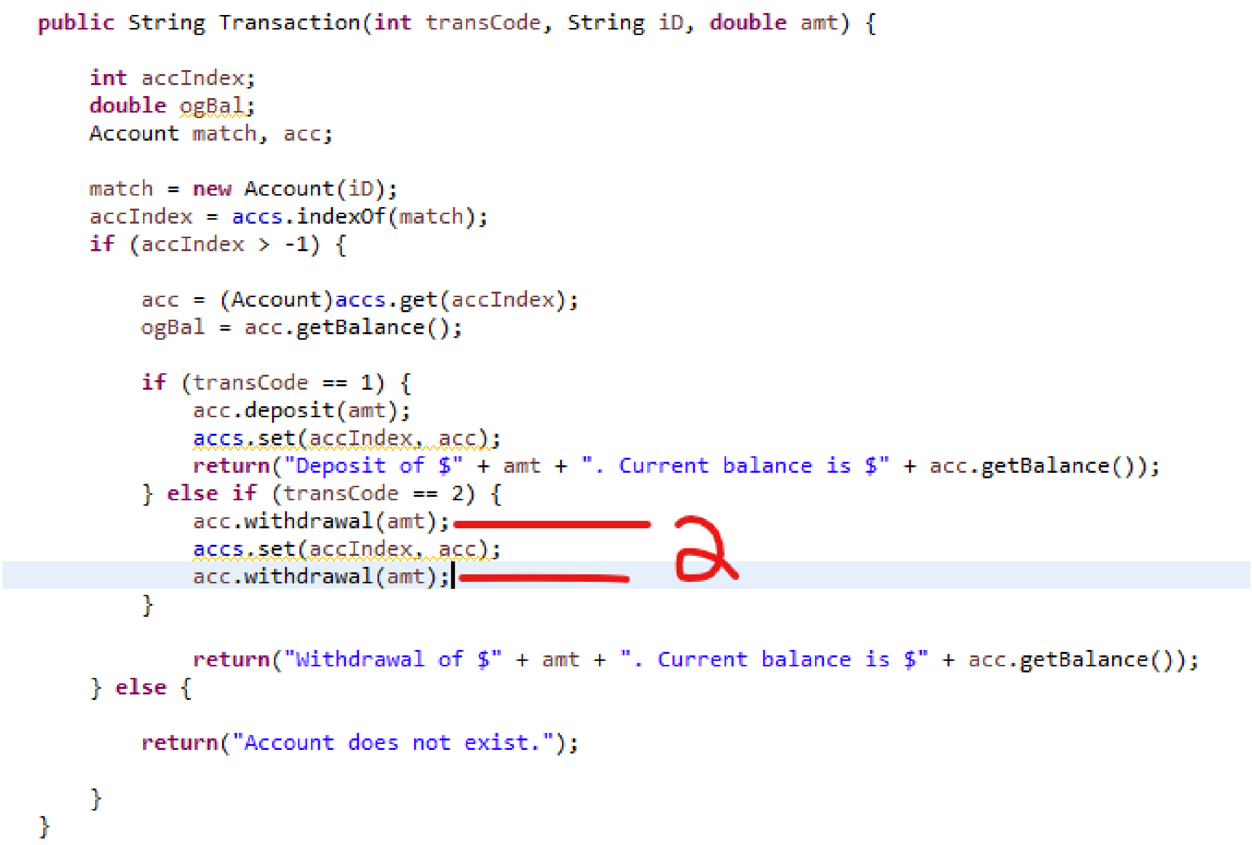
Edit textboxes n... to red labels

| | |
|---|---|
| First name | First Name |
| Last name | Last Name |
| Enter account ID | ID |
| Account Balance | Balance |
| Withdraw Amount | Withdraw |
| Deposit Amount | Deposit |

**Solution** - Set the foreground of every other label that shouldn't be red to black.



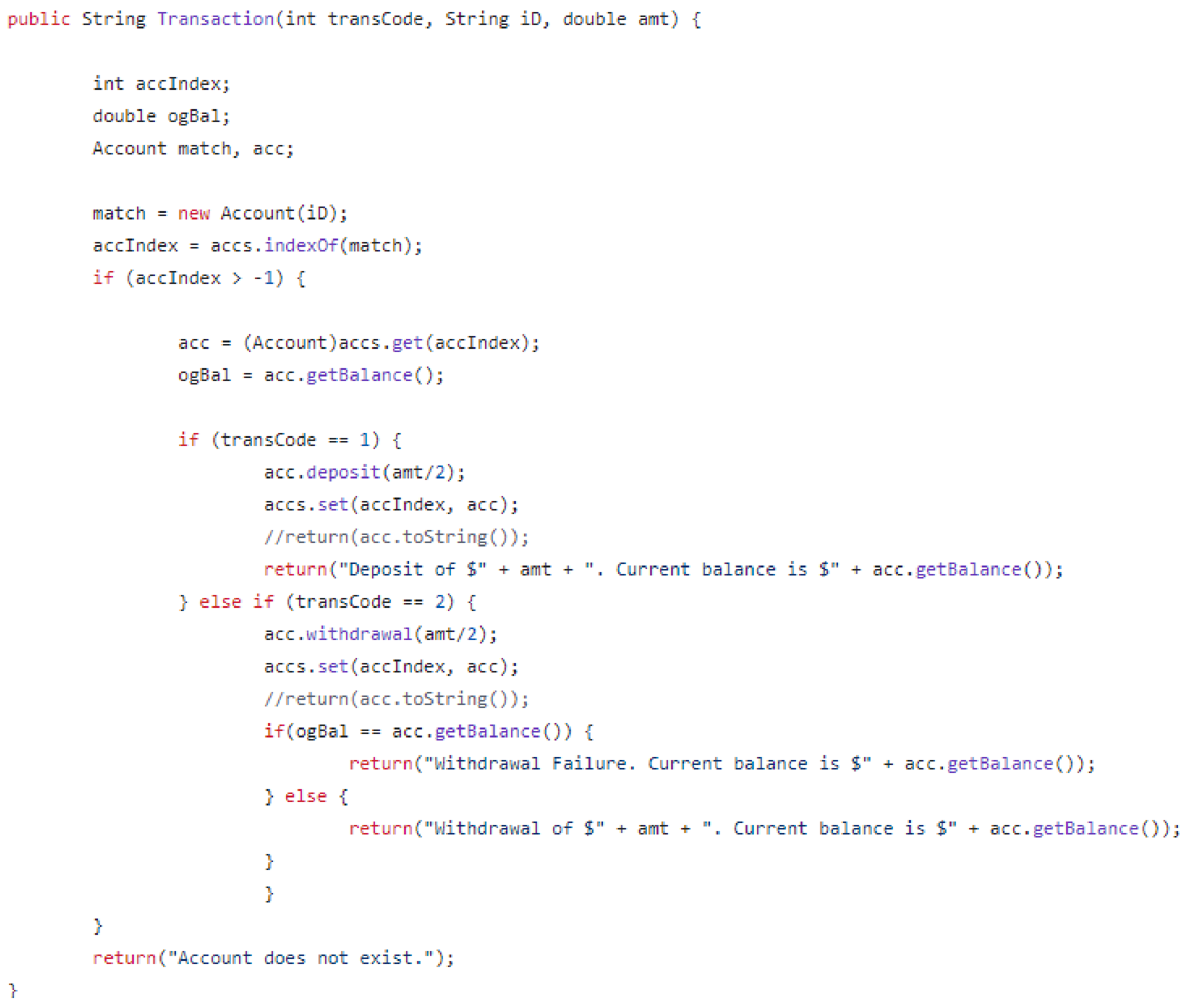
```
if(action.getSelectedItem().equals("Add Account"))
{
    fNLabel.setForeground(Color.red);
    lNLabel.setForeground(Color.red);
    balLabel.setForeground(Color.red);

    iDLabel.setForeground(Color.black);
    depLabel.setForeground(Color.black);
    withLabel.setForeground(Color.black);
}

else if(action.getSelectedItem().equals("Remove Account"))
{

    iDLabel.setForeground(Color.red);

    fNLabel.setForeground(Color.black);
    lNLabel.setForeground(Color.black);
    balLabel.setForeground(Color.black);
    depLabel.setForeground(Color.black);
    withLabel.setForeground(Color.black);
}

else if(action.getSelectedItem().equals("Deposit"))
{

    iDLabel.setForeground(Color.red);
    depLabel.setForeground(Color.red);

    fNLabel.setForeground(Color.black);
    lNLabel.setForeground(Color.black);
    balLabel.setForeground(Color.black);
    withLabel.setForeground(Color.black);

}

else if(action.getSelectedItem().equals("Withdrawal"))
{

    iDLabel.setForeground(Color.red);
    withLabel.setForeground(Color.red);

    fNLabel.setForeground(Color.black);
    lNLabel.setForeground(Color.black);
    balLabel.setForeground(Color.black);
    depLabel.setForeground(Color.black);
```



```java
public String Transaction(int transCode, String iD, double amt) {

    int accIndex;
    double ogBal;
    Account match, acc;

    match = new Account(iD);
    accIndex = accs.indexOf(match);
    if (accIndex > -1) {

        acc = (Account)accs.get(accIndex);
        ogBal = acc.getBalance();

        if (transCode == 1) {
            acc.deposit(amt);
            accs.set(accIndex, acc);
            return("Deposit of $" + amt + ". Current balance is $" + acc.getBalance());
        } else if (transCode == 2) {
            acc.withdrawal(amt);
            accs.set(accIndex, acc);
            acc.withdrawal(amt);
        }

            return("Withdrawal of $" + amt + ". Current balance is $" + acc.getBalance());
    } else {

        return("Account does not exist.");

    }
}
```

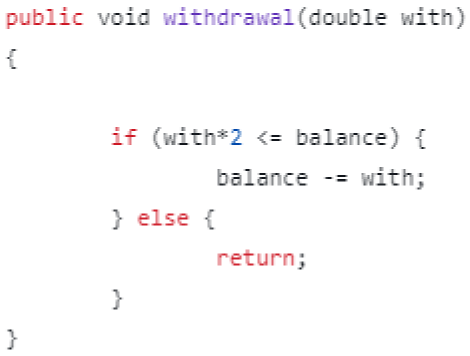
**Error** - Withdrawal wouldn't work if the attempted withdrawal was between 0.5x and 1.5x the current balance. Instead it would withdraw half of the attempted withdrawal. Any other number would function normally

<u>Bank Class</u>



```
public String Transaction(int transCode, String iD, double amt) {

        int accIndex;
        double ogBal;
        Account match, acc;

        match = new Account(iD);
        accIndex = accs.indexOf(match);
        if (accIndex > -1) {

                acc = (Account)accs.get(accIndex);
                ogBal = acc.getBalance();

                if (transCode == 1) {
                        acc.deposit(amt/2);
                        accs.set(accIndex, acc);
                        //return(acc.toString());
                        return("Deposit of $" + amt + ". Current balance is $" + acc.getBalance());
                } else if (transCode == 2) {
                        acc.withdrawal(amt/2);
                        accs.set(accIndex, acc);
                        //return(acc.toString());
                        if(ogBal == acc.getBalance()) {
                                return("Withdrawal Failure. Current balance is $" + acc.getBalance());
                        } else {
                                return("Withdrawal of $" + amt + ". Current balance is $" + acc.getBalance());
                        }
                        }
        }
        return("Account does not exist.");
}
```

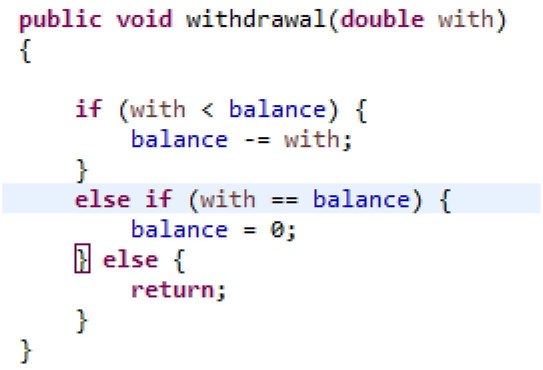
Account Class



```
public void withdrawal(double with)
{

        if (with*2 <= balance) {
                balance -= with;
        } else {
                return;
        }
}
```

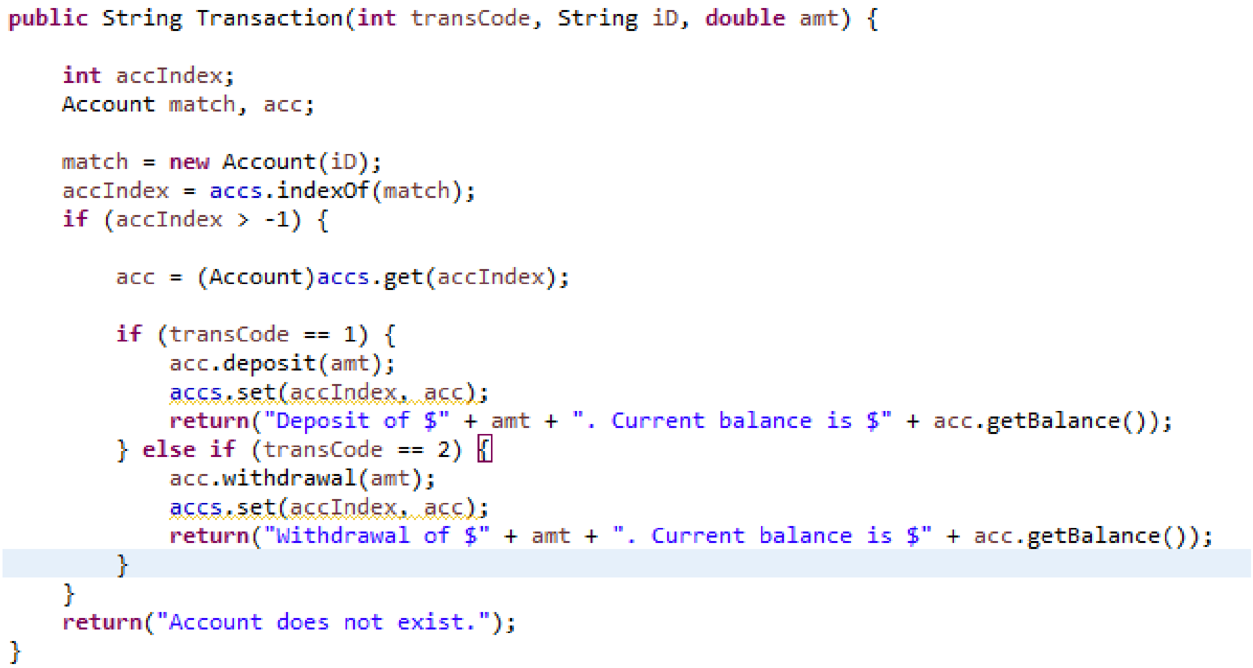
**Solution** - This fixed the error.

Account Class



```java
public void withdrawal(double with)
{

    if (with < balance) {
        balance -= with;
    }
    else if (with == balance) {
        balance = 0;
    } else {
        return;
    }
}
```

Bank Class



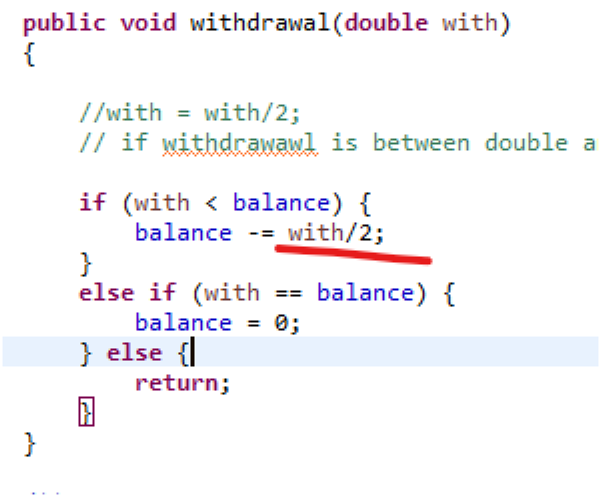
```java
public String Transaction(int transCode, String iD, double amt) {

    int accIndex;
    Account match, acc;

    match = new Account(iD);
    accIndex = accs.indexOf(match);
    if (accIndex > -1) {

        acc = (Account)accs.get(accIndex);

        if (transCode == 1) {
            acc.deposit(amt);
            accs.set(accIndex, acc);
            return("Deposit of $" + amt + ". Current balance is $" + acc.getBalance());
        } else if (transCode == 2) {
            acc.withdrawal(amt);
            accs.set(accIndex, acc);
            return("Withdrawal of $" + amt + ". Current balance is $" + acc.getBalance());
        }
    }
    return("Account does not exist.");
}
```

**Error** - if the withdrawal is between 1x ish - 2x the balance then withdrawal will withdraw half the value.

For example

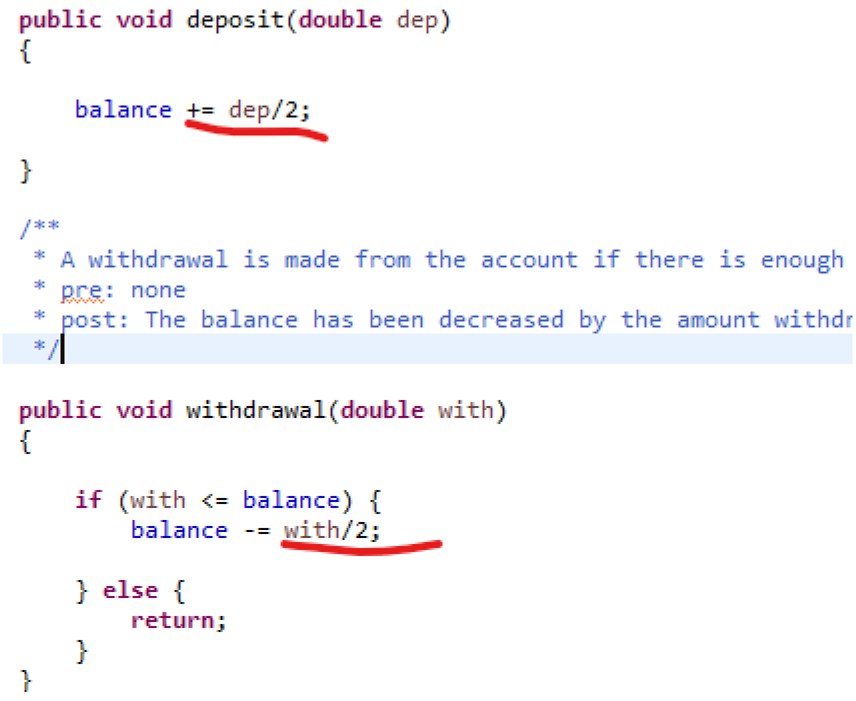
Balance: 100

If the withdraw, variable w, attempt is between 101-200 then the withdrawal will follow through as w/2.

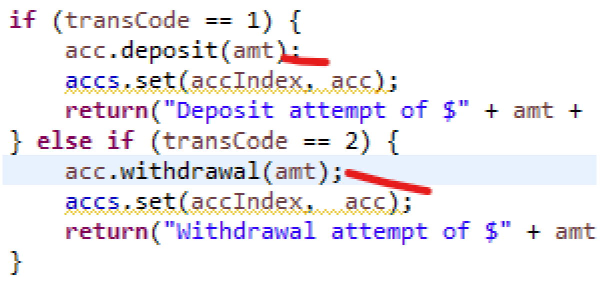


```
public void withdrawal(double with)
{

    //with = with/2;
    // if withdrawawl is between double a

    if (with < balance) {
        balance -= with/2;
    }
    else if (with == balance) {
        balance = 0;
    } else {
        return;
    }
}
...
```

**Solution** - Had to compare the balance to the original value of the withdrawal and divided the withdrawal and deposit by 2 when doing calculations.



```
public void deposit(double dep)
{

    balance += dep/2;

}

/**
 * A withdrawal is made from the account if there is enough
 * pre: none
 * post: The balance has been decreased by the amount withdr
 */

public void withdrawal(double with)
{

    if (with <= balance) {
        balance -= with/2;

    } else {
        return;
    }
}
```



```
if (transCode == 1) {
    acc.deposit(amt);
    accs.set(accIndex, acc);
    return("Deposit attempt of $" + amt +
} else if (transCode == 2) {
    acc.withdrawal(amt);
    accs.set(accIndex,  acc);
    return("Withdrawal attempt of $" + amt
}
```