ATM.java
```
package com.collabra.atm;

import java.io.IOException;

public class Atm extends ATMTransaction {


     public static void main(String[]  args) throws IOException{
           ATMTransaction atmtrans = new ATMTransaction();

          atmtrans.getLogin();
     }

}
```

Account.java
```
package com.collabra.atm;

import java.text.DecimalFormat;
import java.util.Scanner;

public class Account
{
private double checkingBalance =  10000;
private double savingBalance =  10000;
private int customerNumber;
private int pinNumber;

Scanner input = new Scanner(System.in);
DecimalFormat moneyFormat = new DecimalFormat("'$' ###,##0.00");

public int setCustomerNumber(int customerNumber) {
this.customerNumber = customerNumber;
return customerNumber;
}

public int getCustomerNumber() {
     return customerNumber;

}

public int setPinNumber(int pinNumber) {
     this.pinNumber = pinNumber;
     return pinNumber;
}
public int getPinNumber() {
     return pinNumber;

}
```

```
public double getCheckingBalance() {
      return checkingBalance;
}



public double getSavingBalance() {
      return savingBalance;
}

public double calcCheckingWithdraw(double amount) {
      checkingBalance = (checkingBalance - amount);
      return checkingBalance;
}
public double calcSavingWithdraw(double amount) {
      savingBalance = (savingBalance - amount);
      return savingBalance;
}
public double calcCheckingDeposit(double amount) {
      checkingBalance = (checkingBalance + amount);
      return checkingBalance;
}

public double calcSavingDeposit(double amount) {
      savingBalance = (savingBalance + amount);
      return savingBalance;
}

public void getCheckingWithdrawInput() {
      System.out.println("Checking Account Balance:" +
moneyFormat.format(checkingBalance));
      System.out.println("Amount you want to withdraw from Checking
Account:");
      double amount = input.nextDouble();

      if ((checkingBalance - amount) >=0) {
            calcCheckingWithdraw(amount);
            System.out.println("New Checking Account Balance:" +
moneyFormat.format(checkingBalance));

      }
      else {
            System.out.println("Balance cannot be Negative." +"\n");
      }
}

public void getsavingWithdrawInput() {
      System.out.println("Saving Account Balance:" +
moneyFormat.format(savingBalance));
      System.out.println("Amount you want to withdraw from Saving
Account:");
      double amount = input.nextDouble();
```

```
	if ((savingBalance - amount) >=0) {
		calcSavingWithdraw(amount);
		System.out.println("New Checking Account Balance:" +
moneyFormat.format(savingBalance));

	}
	else {
		System.out.println("Balance cannot be Negative." +"\n");
	}
}

public void getCheckingDepositInput() {
	System.out.println("Checking Account Balance:" +
moneyFormat.format(checkingBalance));
	System.out.println("Amount you want to Deposite from Checking
Account:");
	double amount = input.nextDouble();

	if ((checkingBalance + amount) >=0) {
		calcCheckingDeposit(amount);
		System.out.println("New Checking Account Balance:" +
moneyFormat.format(checkingBalance));

	}
	else {
		System.out.println("Balance cannot be Negative." +"\n");
	}
}

public void getSavingDepositInput() {
	System.out.println("Saving Account Balance:" +
moneyFormat.format(savingBalance));
	System.out.println("Amount you want to Deposite from Saving
Account:");
	double amount = input.nextDouble();

	if ((savingBalance + amount) >=0) {
		calcSavingDeposit(amount);
		System.out.println("New Saving Account Balance:" +
moneyFormat.format(savingBalance));

	}
	else {
		System.out.println("Balance cannot be Negative." +"\n");
	}
}
```

ATMTransaction.java

```
} package com.collabra.atm;

import java.io.IOException;
import java.text.DecimalFormat;
import java.util.HashMap;
import java.util.Scanner;

public class ATMTransaction extends Account{

      Scanner menuInput = new Scanner(System.in);
            DecimalFormat moneyFormat = new
DecimalFormat("'$'###,##0.00");

            HashMap<Integer, Integer> data = new HashMap<Integer,
Integer>();

            public void getLogin() throws IOException {
                  int x=1;
                  do {
                        try {
                              data.put(8045264, 3456);
                              data.put(9876548, 7654);
                              System.out.println("Welcome to the ATM");
                              System.out.println("Enter Account Number");
                              setCustomerNumber(menuInput.nextInt());

                              System.out.println("Enter ATM Pin number");
                              setPinNumber(menuInput.nextInt());
                        }
                        catch (Exception e) {
                              System.out.println("\n" + "Invalid
characters. Only numbers" + "\n" );
                              x=2;
                        }

                        int cn =getCustomerNumber();
                        int pn = getPinNumber();
                        if (data.containsKey(cn) && data.get(cn) == pn) {
                              getAccountType();
                        } else
                              System.out.println("\n" + "Wrong Account
number or ATM pin number" + "\n");

                  } while (x==1);
            }

            public void getAccountType() {
                  System.out.println("Select account you want to 
access:");
                  System.out.println("Type 1 - Checking Account");
                  System.out.println("Type 2 - Saving Account");
                  System.out.println("Type 3 - Exit");

                  int selection = menuInput.nextInt();

                  switch (selection) {
                  case 1:
                        getChecking();
                        break;

                  case 2:
                        getSaving();
                        break;

                  case 3:
                        System.out.println("Thank you for using the ATM 
\n");
                        break;

                        default:
                              System.out.println("\n" + "Invalid Choice" 
+ "\n");
                              getAccountType();

                  }
            }



                  public void getChecking() {
                       System.out.println("Checking Account: ");
                       System.out.println("Type 1 - View Balance");
                       System.out.println("Type 2 - Withdraw Amount");
                       System.out.println("Type 3 - Deposit Amount");
                       System.out.println("Type 4 - Exit");
                       System.out.println("Choice: ");

                       int selection1 = menuInput.nextInt();

                       switch (selection1) {
                       case 1:
                            System.out.println("Checking Account 
Balance:" + moneyFormat.format(getCheckingBalance()));
                            getAccountType();
                            break;

                       case 2:
                            getCheckingWithdrawInput();
                            getAccountType();
```

```
                              break;

                         case 3:
                              getCheckingDepositInput();
                              getAccountType();
                              break;
                         case 4:
                              System.out.println("Thank you for using 
ATM");
                              break;

                              default:
                              System.out.println("\n" + "Invalid Choice" 
+ "\n");
                              getChecking();
                         }
                    }



                    public void getSaving() {
                         System.out.println("Saving Account: ");
                         System.out.println("Type 1 - View Balance");
                         System.out.println("Type 2 - Withdraw Amount");
                         System.out.println("Type 3 - Deposit Amount");
                         System.out.println("Type 4 - Exit");
                         System.out.println("Choice: ");

                         int selection2 = menuInput.nextInt();

                         switch (selection2) {
                         case 1:
                              System.out.println("Checking Account 
Balance:" + moneyFormat.format(getSavingBalance()));
                              getAccountType();
                              break;

                         case 2:
                              getsavingWithdrawInput();
                              getAccountType();
                              break;

                         case 3:
                              getSavingDepositInput();
                              getAccountType();
                              break;
                         case 4:
                              System.out.println("Thank you for using 
ATM");
                              break;

                         default:
```

```
                                        System.out.println("\n" + "Invalid Choice"
+ "\n");
                                        getSaving();

                                }

                        }
}
```