COMP 10182

Assignment #5

**Topics Covered:** MODULES 1 to 8  
**Weight:** 10%  
**Submission Instructions:** Download the provided Excel file and JavaScript file and create an HTML file. Name the files using your student number for each. They should contain the requirements of the assignment. When your assignment is complete, add a second .txt file extension to the HTML and JavaScript files or they may be blocked from upload. For example:

123456789.html.txt

123456789.js.txt  
 123456789.xlsx

Upload your files to the eLearn Dropbox. Check to see if there is a quiz to streamline the marking of the assignment. If there is a quiz, complete the quiz.

**Instructions:**

1. Follow the style guide best practices for JavaScript.
2. Provide your name, student number, date created, and program description at the beginning of the JavaScript file using multiline comments or a series of single line comments. You should comment the beginning of any significant code blocks with important functionality with a description of what is expected.

Use your JavaScript and HTML knowledge acquired from the course to write a program that conducts the following.

1. Write an ABM (automated banking machine) program that takes input from the user:
   1. Asks for card number and compares it to a value in an array
   2. Asks for pin and compares it to a corresponding value in an array
   3. Displays a set of options and asks for a transaction type
      1. Make a deposit
      2. Withdraw cash
      3. Check balance
   4. Selects an account
2. The following output should be made:
   1. Updated account balance after transaction.
   2. Allows the user to conduct more transactions.
3. Properly comment all files created.
4. Use any appropriate input/output technique you learned in the course (prompts, alerts, or DOM HTML updates).
5. Create any variables and initial values such as account amounts.
6. Try to code for exceptions that may break the system and show alerts to the user if they occur. For example, what if a pin doesn’t match? What if they try to withdraw more than they have on file?
7. Create and/or link as many JavaScript files as necessary with the HTML file.
8. Submit your completed files using the instructions above.

// withdrawal

if(choice.equals("wi")){

System.out.print("Enter amount to be withdrawn: ");

double amount = scan.nextDouble();

if (amount > account.balance()){

System.out.println("You can't withdraw that amount!");

} else if (amount <= account.balance()) {

account.withdraw(amount);

System.out.println("NewBalance = " + account.getBalance());

}

// deposit

if(choice.equals("dp")){

System.out.print("Enter amount to deposit: ");

double amount = scan.nextDouble();

if(amount <= 0){

System.out.println("You must deposit an amount greater than 0.");

} else {

System.out.println("You have deposited " + (amount + account.getBalance()));

}

// account selection

if(choice.equals("se")){

System.out.println("Enter number of account to be selected: ");

//user input for account nr from array

System.out.println("Account closed.");

public class Account {

private Integer accountNumber;

private Double balance;

public Account(final Integer accountNumber, final Double initialBalance) {

this.accountNumber = accountNumber;

balance = initialBalance;

}

public Double deposit (double depositAmmount) {

balance += depositAmmount;

return balance;

}

public Double withdraw(double withdrawAmmount) {

balance -= withdrawAmmount;

return balance;

}

public Double getBalance() {

return balance;

}

public Integer getAccountNumber() {

return accountNumber;

}

}

//select account

else if(choice.equals("se")) {

System.out.println("Enter number of account to be selected: ");

Integer accountToGetNumber = input.nextInt();

Account returnedAccount = accountMap.get(accountToGetNumber);

if (returnedAccount != null)

{

System.out.println("Account open. Current balance: " + returnedAccount.getBalance());

}

else

{

//user input for account nr from array

System.out.println("Account does not exist.");

}

}

//close account

else if(choice.equals("de"))

{

System.out.println("Enter number of account to be selected: ");

Integer accountToDeleteNumber = input.nextInt();

Account removedAccount = accountMap.remove(accountToDeleteNumber);

if (removedAccount != null)

{

System.out.println("Account " + removedAccount.getAccountNumber() + " has been closed with balance: " + removedAccount.getBalance());

}

else

{

//user input for account nr from array

System.out.println("Account does not exist.");

}

}

// deposit

else if(choice.equals("dp")) {

System.out.println("Enter number of account to deposit: ");

Integer accountToDeposit = input.nextInt();

System.out.print("Enter amount to deposit: ");

double amount = input.nextDouble();

if(amount <= 0){

System.out.println("You must deposit an amount greater than 0.");

} else {

accountMap.get(accountToDeposit).deposit(amount);

System.out.println("You have deposited " + (amount));

System.out.println("Current balance " + accountMap.get(accountToDeposit).getBalance());

}

}

// withdrawal

else if(choice.equals("wi")) {

System.out.println("Enter number of account to withdraw: ");

Integer accountToWithdraw = input.nextInt();

System.out.print("Enter amount to withdraw: ");

double amount = input.nextDouble();

if(amount <= 0) {

System.out.println("You must deposit an amount greater than 0.");

} else {

accountMap.get(accountToWithdraw).withdraw(amount);

System.out.println("You have deposited " + (amount));

System.out.println("Current balance " + accountMap.get(accountToWithdraw).getBalance());

}

}

//quit

else if(choice.equals("ex")) {

break;

} else {

System.out.println("Wrong option.");

} //end of if

} //end of loop

input.close()