CS 150 Lab 7
Data Validation & Menus

The main objective of today's lab is to use the do/while loop to validate data as well as use a menu to help the user navigate a program.

Be sure your output looks exactly like the specified output.
Be sure to submit the completed project to CS150-02 Lab by Friday at 5pm.

Show the instructor or TA your solution after EACH screen is implemented below.

Lab 7.1

Banks have to process many transactions for many customers every month. Write a C++ program in a project **7_1_BankTransactions** to process one transaction. A nested loop is required to process multiple transactions. We haven't yet talked about nested loops.

Here is how your program is to work:

Screen #1:

```
****************************************
*        Pacific Credit Union          *
****************************************

Enter Beginning Balance: $ 1000.00
```

Screen #2:

```
****************************************
*        Pacific Credit Union          *
****************************************

W)ithdrawal
D)eposit
B)alance

Select: W
```

Screen #3:

```
****************************************
*        Pacific Credit Union          *
****************************************

Enter the withdrawal amount $ 100.00

Account Balance: $ 900.00
```

1) You are to validate all input as follows:

a) Monetary input is to be greater than zero

b) The user selection is to be either W, D, or B

c) Do not let the user withdraw more money than is in the account

2) Use as few variables as possible.

Note1: If any user input is invalid, simply clear the screen with the C++ statement **system ("cls");** and then repeat the screen until the user enters valid data.

Note2: Always clear the screen before continuing on to the next screen.

Challenge1: If the user enters invalid input, display an appropriate error message for about 2 seconds before clearing the screen and then displaying the same exact input screen again.

Challenge2: Modify the above program to allow unlimited deposits and withdrawals until the user enters B at which time your program is to display the ending balance.

Challenge3: Print the minimum and maximum account balance as a summary statistic.

1) Your program is to compile without any errors or warnings.

2) Do not use any magic constants in your program. Define your constants before defining the rest of your program's variables.

Once your project is complete, place your solution PUNetIDLabs into the CS150-02 Drop folder on Turing. Your solution is to have all previous projects completely working and correct.