CS 215 – Fall 2018
Project 1: ATM

Learning Objectives: Use of the following to solve a problem:

- C++ basic data types, variables, assignments, arithmetic
- user input/output
- decisions
- basic loops

General Description:

Write a C++ program that simulates the function of an ATM machine. For each transaction, the program:

- asks the user to enter their current account number and balance (unrealistic, but we don't know how to read data from files yet).
- displays a menu of transaction options
- asks the user to choose an option, forcing them to enter a correct option
- depending on the option chosen, asks for more information
- performs the calculations for the transaction
- prints the calculated amounts as a "receipt".
-

The program repeats the above steps continuously until the user enters **shutdown** for the account number. This is a signal for the ATM to shut down (end program).

Specifications (*example* user input in blue):

Beginning of transaction:

```
+---------------------+
| Joiner's Bank, Inc. |
|      CS215-003      |
+---------------------+
Enter account number:  BA-3456-78
Enter account balance: 275.50
```

Be creative with the "Logo"
*use your **name** and **section** somewhere in the Logo*

these two entries should "line up" on the left as shown.

OR, if they enter **shutdown**

```
+---------------------+
| Joiner's Bank, Inc. |
|      CS215-003      |
+---------------------+
Enter account number:  shutdown
Shutting down...bye!
Press any key to continue...
```

use the standard System Pause at end of program

Menu: after printing the options, it should repeatedly ask for an option until the user enters B, D or W.

```
B - Balance Inquiry
D - Deposit
W - Withdrawal
Choose an option:  Xxx
"Xxx" is an invalid option. Enter B, D or W.
Choose an option:  b
"b" is an invalid option. Enter B, D or W.
Choose an option:  B
```

assume they might enter more than one char
print double quotes around any invalid entry
It does not need to accept both upper and lower case
Finally! They gave a correct answer

Balance Inquiry: when this option is chosen, just print a simple "receipt":

```
+---------------------+
| Joiner's Bank, Inc. |
+---------------------+
Account:   BA-3456-78          should line up with the dollar sign below
Balance:   $  275.50           Assume the max balance is 99999.99 5 digits before, 2 after the point.
```

Deposit: when this option is chosen, ask for the deposit amount (assume they will enter a float) and print a receipt:

```
Enter deposit amount: 100.00
+---------------------+
| Joiner's Bank, Inc. |
+---------------------+
Account:   BA-3456-78          All should line up as shown.
Prev Bal:  $  275.50           All amounts assumed to be 99999.99 or less.
Deposit:   $  100.00
New Bal:   $  375.50
```

Withdrawal: when this option is chosen, ask for the withdrawal amount (assume they will enter a float). Repeat the question as long as the amount entered is greater than the current balance. Then print a receipt:

```
Enter withdrawal amount: 500.00
Insufficient Funds. Current Balance is $ 275.50
Enter withdrawal amount: 100.00
+---------------------+
| Joiner's Bank, Inc. |
+---------------------+
Account:   BA-3456-78          All should line up as shown.
Prev Bal:  $  275.50           All amounts assumed to be 99999.99 or less.
Withdrawn: $  100.00
New Bal:   $  175.50
```

Repeat: after the transaction is complete and the receipt printed, the program should start over at the beginning (printing the logo) for the next customer.

Suggestion:

Since we won't be covering loops until next week, you can go ahead and start by coding the project assuming:

- it won't repeat for multiple customers; do one customer and end program
- it won't repeat for an invalid menu option; assume B was entered and continue (You can still print the error message)
- It won't repeat for insufficient funds; just let them have a negative balance. (You can still print the error message)

Later, after you learn about loops, you can go back and *add* the loops to this "mostly done" project.

Style:
10% of your grade will be based on Style. Follow these rules:
1. At the top, put a comment box with the following info. This is required for all projects:
- a program title, centered at the top
- Name
- Date
- "CS215 Section " nnn
- A brief description of the program

2. Use indentation inside of **blocks { }** and control statements as specified in class (and suggested by Visual Studio's editor)

3. Use meaningful variable names.

4. Comment sections of the program with a blank line and a comment (lined up with the code). The "sections" listed in the Specifications above can be used to break the code up into sections. Ex:

```
void main() {

      //transaction begin
      . . .
      . . .

      // print menu
      . . .
      . . .

      // get user selection
      . . .
      . . .
}
```

5. If needed for clarity, place comments on the end of individual lines.

Submission:
- Submit your Source (.cpp) File in Canvas.
- You do NOT need to zip up and submit an entire folder.
- You do NOT need to demonstrate projects to the TA…the date/time of submission in Canvas is your official submission time.