CSE 1321L: Programming and Problem Solving I Lab

Assignment 2 Fall 2025

Module 2

What students will learn:

- Reinforcing the concept of Boolean expressions and conditionals.
- Implementing selection statements within repetition structures (loops).

Content

- Overview
- Assignment2A: OwlBanking ATM
- Assignment2B: Escape Room
- Assignment2C: Temperature Average

Overview:

For this assignment, you are going to practice implementing selection structures within repetition structures. Thus, reinforcing the use of selection statements but also learning how to create more complex problems by adding repetition into your programs.

Final note: ***Do not cheat***
If your temptation is to look online, don't. Come see us instead and ask questions – we are here to help. Remember, you are going to have to write code in your future job interviews, so learn it now to secure a high-paying job later.

Assignment2A: OwlBanking ATM

For this assignment, you are going to build a program that behaves like an Automatic Teller Machine or ATM. With your program, the user should be able to deposit, withdraw, and check their balance. Also, the ATM should feature a transaction history tracker that the user should be able to review.

Requirements

- The user should be able to keep interacting with the program until the user decides to quit. To do so, your program must implement a **while** loop.
- Display these options to the user:
    - 1 – Deposit
    - 2 – Withdraw
    - 3 – Check Balance
    - 4 – Check Transaction History
    - Q – Quit
- Your implementation must use any of the selection structures covered in the first part of Module 2.
- The user will start with a balance of $0 dollars.
- If the user chooses to deposit, prompt the user to enter how much they want to deposit and **add** it to the user's balance.
- If the user chooses to withdraw:
    - Check if the balance is $0. If so, give the user an error message and **do not** let the user withdraw.
    - If the balance is greater than $0, then ask how much to withdraw.
    - The balance cannot drop lower than $0, if the amount to withdraw will make the balance go lower than $0 give the user an error message and **do not** let the user withdraw.
    - If the balance does not drop lower than $0, subtract the withdrawal amount from the balance.
- If the user chooses to check the balance, output the current balance.
- If the user chooses to check transaction history, output the transaction history.
    - Any deposit or withdrawal operation **must** be logged. After each successful deposit or withdrawal, make sure the program is recording each operation.
- If the users chooses option 'Q' stop the program.
- Do follow the sample output format provided as a guide on what to output and how to format your print statements.

Hint

- Your program must log and remember each deposit and withdrawal. We have not covered sequences yet such as list but remember that you can still use strings and string concatenation to store this information.

Sample Output #1:

```
[Welcome to OwlBanking]
Select an option:
1 - Deposit
2 - Withdraw
```

```
3 - Check Balance
4 - Check Transaction History
Q - Quit
2

Error: Cannot Withdraw with balance of zero
Select an option:
1 - Deposit
2 - Withdraw
3 - Check Balance
4 - Check Transaction History
Q - Quit
3

Balance: $0.00
Select an option:
1 - Deposit
2 - Withdraw
3 - Check Balance
4 - Check Transaction History
Q - Quit
1

How much do you want to deposit: $1000
Select an option:
1 - Deposit
2 - Withdraw
3 - Check Balance
4 - Check Transaction History
Q - Quit
3

Balance: $1000.00
Select an option:
1 - Deposit
2 - Withdraw
3 - Check Balance
4 - Check Transaction History
Q - Quit
2

How much do you want to withdraw: $500
Select an option:
1 - Deposit
2 - Withdraw
3 - Check Balance
4 - Check Transaction History
Q - Quit
4
```

```
$1000.00 was deposited, balance went from $0.00 to $1000.00
$500.00 was withdrawn, balance went from $1000.00 to $500.00

Select an option:
1 - Deposit
2 - Withdraw
3 - Check Balance
4 - Check Transaction History
Q - Quit
3

Balance: $500.00
Select an option:
1 - Deposit
2 - Withdraw
3 - Check Balance
4 - Check Transaction History
Q - Quit
2

How much do you want to withdraw: $501
Error: Cannot Withdraw a larger amount than balance
Select an option:
1 - Deposit
2 - Withdraw
3 - Check Balance
4 - Check Transaction History
Q - Quit
3

Balance: $500.00
Select an option:
1 - Deposit
2 - Withdraw
3 - Check Balance
4 - Check Transaction History
Q - Quit
2

How much do you want to withdraw: $499
Select an option:
1 - Deposit
2 - Withdraw
3 - Check Balance
4 - Check Transaction History
Q - Quit
3

Balance: $1.00
Select an option:
```

```
1 - Deposit
2 - Withdraw
3 - Check Balance
4 - Check Transaction History
Q - Quit
4

$1000.00 was deposited, balance went from $0.00 to $1000.00
$500.00 was withdrawn, balance went from $1000.00 to $500.00
$499.00 was withdrawn, balance went from $500.00 to $1.00

Select an option:
1 - Deposit
2 - Withdraw
3 - Check Balance
4 - Check Transaction History
Q - Quit
Q
```

Assignment 2B: Escape Room

For this assignment, you are going to build a simple escape room game. Please do stick with the story and game logic explained here, **do not** create your own.

The game should start with this message:
"You wake up in a dimly lit room. The air smells faintly of dust and old wood. The only visible exit is a heavy iron door with a large lock. Looking around, you notice:

- A small wooden chest sitting on a table in the corner
- A bare mattress on the floor
- A section of the floorboards that looks slightly uneven, almost as if one plane doesn't quite fit"

The game will consist of 4 options the user can choose to progress:

A. Open the heavy iron door.
B. Inspect the wooden chest.
C. Inspect the mattress.
D. Inspect the suspicious plank in the floorboard.

To escape, the user must:

1. Find the brass key under the suspicious plank.
2. Use the brass key on the wooden chest to reveal the iron key.
3. Use the iron key to open the heavy iron door.

The game will display a different message depending on the user's previous action and depending on the user's progress in the game. Here is a list of messages your game should display:

Messages for action A:

- If the user tries to open the heavy iron door without having found the iron key:
    - "You tug the handle. It doesn't budge. The key must be somewhere in this room…"
- If the user has found the brass key:
    - "The brass key is too small for the door's lock. Maybe it opens something else."
- If the user had found the iron key:
    - "The iron key slides into the lock. You turn it, the mechanism clicks. The door swings open. You're free!"

Messages for action B:

- If the users inspect the wooden chest without the brass key and without the iron key:
    - "The chest is locked. The key has to be around here somewhere…"
- If the user has found the brass key:
    - "The brass key fits the chest lock. It clicks open. Inside rests a heavier iron key. This one looks it is made for a door."
- If the user has found both brass and iron key:
    - "You check the chest again, nothing else inside."

Messages for action C:

- If the user has not interacted with the mattress:

    - "You inspect the thin mattress, there is nothing on it. You lift it, there is nothing underneath either."
- If the user has interacted with the mattress:
    - "You already checked the mattress. There is nothing there."

Messages for action D:
- If the user has not found the brass key:
    - "You pry up the loose plank. Hidden in the dust lies a small brass key."
- If the user has found the brass key:
    - "You've already pried up the plank. There is nothing else underneath."

Requirements

- The user should be able to keep playing the game user finds all keys and escapes. To do so, your program must implement a **while** loop.
- Your program should follow the logic previously discussed.
- Each action should only print one message based on the user's previous interactions.
    - For example, if the user has already found the brass key under the suspicious plank and decides to check again, the game should print "You've already pried up the plank. There is nothing else underneath."
- Do follow the sample output format provided and the game logic explained above.

Hint

- You could track if the user has done or not done an action with a Boolean.

Sample Output

```
You wake up in a dimly lit room. The air smells faintly of dust and old wood.
The only visible exit is a heavy iron door with a large lock.
Looking around, you notice:
- A small wooden chest sitting on a table in the corner
- A bare mattress on the floor
- A section of the floorboards that looks slightly uneven, almost as if one
plank doesn't quite fit.

What do you do?
A. Open the heavy iron door.
B. Inspect the wooden chest.
C. Inspect the mattress.
D. Inspect the suspicious plank in the floorboard.
A
You tug the handle. It doesn't budge. The key must be somewhere in this
room...

What do you do?
A. Open the heavy iron door.
B. Inspect the wooden chest.
C. Inspect the mattress.
```

D. Inspect the suspicious plank in the floorboard.
C
You inspect the thin mattress, there is nothing on it. You lift it, there is nothing underneath either.

What do you do?
A. Open the heavy iron door.
B. Inspect the wooden chest.
C. Inspect the mattress.
D. Inspect the suspicious plank in the floorboard.
D
You pry up the loose plank. Hidden in the dust lies a small brass key.

What do you do?
A. Open the heavy iron door.
B. Inspect the wooden chest.
C. Inspect the mattress.
D. Inspect the suspicious plank in the floorboard.
C
You already checked the mattress. There is nothing there.

What do you do?
A. Open the heavy iron door.
B. Inspect the wooden chest.
C. Inspect the mattress.
D. Inspect the suspicious plank in the floorboard.
D
You've already pried up the plank. There is nothing else underneath.

What do you do?
A. Open the heavy iron door.
B. Inspect the wooden chest.
C. Inspect the mattress.
D. Inspect the suspicious plank in the floorboard.
B
The brass key fits the chest lock. It clicks open. Inside rests a heavier iron key. This one looks it is made for a door.

What do you do?
A. Open the heavy iron door.
B. Inspect the wooden chest.
C. Inspect the mattress.
D. Inspect the suspicious plank in the floorboard.
B
You check the chest again, nothing else inside.

What do you do?
A. Open the heavy iron door.
B. Inspect the wooden chest.
C. Inspect the mattress.

D. Inspect the suspicious plank in the floorboard.
A
The iron key slides into the lock. You turn it, the mechanism clicks. The door swings open. You're free!

Assignment 2C: Temperature Average

In this assignment, you are going to create a program that calculates the average temperature (in F) given a series of temperatures.

Requirements:

- The user will input a series of multiple temperatures separated with a comma ,
  - This means that the user will input an undetermined amount of temperature values.
- We have not cover sequences yet, therefore you **must** solve this problem by traversing the input string with a **FOR** loop.
  - This means, you will check character by character your string to determine if a character is a number or if it is the delimiter (a comma in our case).
  - This also means that your program should be evaluating if a number is either the beginning or the end of a temperature value (remember that a temperature reading in F usually has 2 digits or more) and **count** them.
- After your program is done processing the temperatures, calculate the average temperature and print it.
  - This means your program should also **count** how many temperature values the user has input.
- Do follow the sample output format provided as a guide on what to output and how to format your print statements.

Hints

- Depending on how you approached the problem, you will have an issue with the last temperature reading. There are a couple of ways you can approach this, you could either perform an extra operation to count the last temperature or maybe you can concatenate an extra comma to the input.
- When you are traversing the user input string, if you find a comma there are two assumptions to be made:
  - We just ended up reading the last digit of the previous temperature value.
  - We are about to start reading the first digit of the next temperature value.

Sample Output #1

```
Enter temperatures in F (separate them with ,): 78,79.5,80,78,81,83
You've entered 6 temperature points.
The average temperature is 79.92F
```

Sample Output #2

```
Enter temperatures in F (separate them with ,): 81.4,81.6,81,80,80.6,79.3,82
You've entered 7 temperature points.
The average temperature is 80.84F
```