COP3331 Lab 4

Submission Instructions:

1. Create a folder named Lab4_lastNamefirstInitial (e.g. Lab4_NealT).
2. In your folder, place a **PDF** file containing your answers to questions with a ♢.
3. Copy your directories containing your programs for questions with a ♠ into the folder; **these directories should only contain files needed to run your program, which may include one or more of the following file types: .cpp, .h., and .txt**. Do NOT include the full project (e.g., solution file). Test your program on CIRCE before submitting by compiling and running with **g++**. Your file containing main() should **always** be named main.cpp.
4. Ensure that all programs have block comments at the very beginning (starting at the first line) in the file containing main() with your name and the program's description. **The block comment's format should be identical to what's provided in Figure 2-1.**
5. Use single-line comments to describe your code's functionality as needed.
6. Do not submit anything for questions with a ♣.
7. Zip the folder and submit it via Canvas.

♢ = 5 points each, ♠ = 15 points each

---

1. ♣ Read *Chapter 9: How to work with structures and enumerations.*
2. ♢ Which of the following statements is not true about structures?
   a. They define a data type.
   b. They are typically used to organize related data.
   c. They include data members that must be fundamental data types.
   d. They can include member functions that operate on the data members.

3. ◇ The advantage of returning a structure type from a function when compared to returning a fundamental type is that
   a. the function can return multiple values
   b. the function can return an object
   c. the function doesn't need to include a return statement
   d. all of the above
   e. a and b only

4. ◇ An enumeration contains enumerators that represent
   a. the variables for the enumeration
   b. the constants for the enumeration
   c. the operators for the enumeration
   d. the functions for the enumeration

5. ◇ What are the values of the enumerators in the enumeration that follows?

```
enum class Terms {
    net_30_days = 30,
    net_60_days,
    net_90_days
};
```

   a. 30, 60, 90
   b. 30, 1, 2
   c. 30, 31, 32
   d. 30, 0, 1

6. ♣ Read *Chapter 10: How to work with STL containers and iterators.*

7. ◇ What are the values of the elements in the vector named names_1 after the following code is executed?

```
vector<string> names_1 { "Mike", "Ben", "Joel", "Anne" };
vector<string> names_2 { "Judy", "Samantha", "Kelly" };

names_1.insert(names_1.end(), "Mary");
names_1.erase(names_1.begin());
names_1.insert(names_1.begin() + 2, ++names_2.begin(), names_2.end());
```

```
names_1.swap(names_2);
names_1.erase(++names_1.begin());
names_1.insert(names_1.begin(), ++names_2.begin(), names_2.begin() + 2);
```

a. Joel, Judy, Kelly
b. Judy, Mary, Joel, Mary
c. Joel, Judy, Samantha
d. Joel, Anne, Judy, Samantha

8. ♢ A stack container provides
a. last-in, first-out access
b. last-in, last-out access
c. first-in, last-out access
d. first-in, first-out access

9. ♢ When you dereference an iterator, you
a. set the value of the iterator variable to null
b. set the iterator so it points one memory location past the last element in the container
c. get the value of the element that the iterator points to
d. get the memory address of the element that the iterator points to

10. ♠ **Program Lab4_10: Menu of the Day**
Create a program that filters and displays the correct menu items based on the day that the user enters. Save in folder `lab4-q10`.

Console

```
Menu of the Day

COMMAND MENU
Specify the day using the three-letter format
(mon, tue, wed, thu, fri, sat, sun).
Or, enter 'exit' to exit.

Day:  mon
Roast beef               10
Potato soup              6
```

```
Baked cod                10

Day:  tue
Butter chicken           11
Lamb tikka               12
German forest cake       7

Day:  exit
Bye!
```

Specifications

- I've provided a tab-delimited text file named menu.txt that contains the menu items for the current week. Each item has a name (a string), price (an integer), and associated day (a string)
- Use a structure named MenuItem to store the name, price, and day for each menu item.
- When the program starts, it should load the items from the file into a vector of MenuItem objects.
- The user should be able to type a three-letter day code to display the menu items for the specified day.
- The user should be able to exit by typing "exit" when prompted for the day.

11. ♠ **Program Lab4_11: Tic Tac Toe (STL version)**
Create a two-player Tic Tac Toe game. Save in folder `lab4-q11`.

Console

```
Welcome to Tic Tac Toe

+---+---+---+
|   |   |   |
+---+---+---+
|   |   |   |
+---+---+---+
|   |   |   |
+---+---+---+

X's turn
Pick a row (1, 2, 3): 1
Pick a column (1, 2, 3): 1

+---+---+---+
| X |   |   |
+---+---+---+
|   |   |   |
+---+---+---+
|   |   |   |
+---+---+---+

O's turn
Pick a row (1, 2, 3): 1
Pick a column (1, 2, 3): 2

...
...

X's turn
Pick a row (1, 2, 3): 3
Pick a column (1, 2, 3): 3

+---+---+---+
| X | O | O |
+---+---+---+
|   | X |   |
+---+---+---+
|   |   | X |
+---+---+---+
X wins!

Game over!
```

Specifications

- Use an array of STL arrays to store the Tic Tac Toe grid.
- If the user picks an invalid row or column or a cell that's already taken, display an error message.
- If there is a winner, the game should display an appropriate message and end. Otherwise, it should continue until the grid is full and end in a tie.