COP3331 Lab 3

Submission Instructions:

1. Create a folder named Lab3_lastNamefirstInitial (e.g. Lab3_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 5: How to work with I/O streams and files.*
2. ♢ A buffer is
   a. a location in memory where stream data is stored after it's read or written
   b. a location on disk where stream data is stored after it's read or written
   c. a location in memory where stream data is stored as it's read or written
   d. a location on disk where stream data is stored as it's read or written

3. ♢ Assume that a program displays two prompts that each accept a value from the user and no data is discarded after each entry. What will happen if the user enters both values at the first prompt?
   a. The program will extract the first value and then prompt for and extract the second value without waiting for the user to enter that value.
   b. The program will extract the first value, ignore the second value, and then prompt for another value.
   c. The program will extract both values as a single value.
   d. The program will ignore both values and end abnormally.

4. ♠ Create a program that reads the sales for 12 months from a file and calculates the total yearly sales as well as the average monthly sales. The user can enter 'm', 'y', or 'x' as commands to view the sales, yearly summary, or exit the program, respectively. Save your program in a folder `lab3_q4`.

Console

```
Monthly Sales

COMMAND MENU
m - View monthly sales
y - View yearly summary
x - Exit program

Command: m

MONTHLY SALES
Jan        14317.41
Feb         3903.32
Mar         1073.01
Apr         3463.28
May         2429.52
Jun         4324.70
Jul         9762.31
Aug        25578.39
Sep         2437.95
Oct         6735.63
Nov          288.11
```

```
Dec         2497.49

Command: a

Invalid command. Try again.

Command: y

YEARLY SUMMARY
Yearly total:           76811.12
Monthly average:         6400.93

Command: x

Bye!
```

Specifications

- I've provide a tab-delimited file named monthly_sales.txt that contains the month and sales data shown above.
- Round the results of the monthly average to a maximum of 2 decimal digits and make sure that a minimum of 2 decimal digits are displayed on the console.
- Right-align the columns that display the monthly sales, the yearly total, and the monthly average.

Note for Xcode users

- To get this to work correctly on your system, you can set a full path to the file in your code, or you can change the working directory for the Xcode project to the directory that contains the text file.
- To change the working directory for the Xcode project, open the project and select the Product→Scheme→Edit Scheme item. Then, select the Run category, check the Use Custom Working Directory box, and specify the working directory.

5. ♣ Read *Chapter 7: How to code functions.*

6. ♠ Create a program in folder `lab3_q6` that uses a separate module to calculate sales tax and total after tax.

Console

```
Sales Tax Calculator

ENTER ITEMS (ENTER 0 TO END)
Cost of item: 35.99
Cost of item: 27.50
Cost of item: 19.59
Cost of item: 0

Total:           83.08
Sales tax:        4.98
Total after tax: 88.06

Again? (y/n): y

ENTER ITEMS (ENTER 0 TO END)
Cost of item: 152.50
Cost of item: 59.80
Cost of item: 0

Total:           212.30
Sales tax:        12.74
Total after tax: 225.04

Again? (y/n): n

Thanks, bye!
```

Specifications

- The program should only accept numbers that are greater than 0.
- Use the console.h and console.cpp files described in chapter 7 and provided via Canvas to validate user entries. That way, the user can't crash the program by entering invalid data.
- The sales tax rate should be 6% of the total.

- Use a header file to declare two functions. One should accept the total of the items and return the tax amount. The other should accept the total of the items and return the total after tax has been added.
- Use the implementation file for this header file to store the sales tax rate and the definitions for these two functions. These functions should round the results to two decimal places.
- The output should display all monetary values with 2 decimal places.
- The output should right align the numbers in the second column. This makes it easier to check whether the calculations are correct.

7. ♢ What happens if you attempt an operation that uses the input stream and the operation fails but the stream is OK?
   a. Any data that remains in the stream is automatically discarded.
   b. The program will end abnormally.
   c. The failbit of the stream object will be set.
   d. All of the above
   e. a and c only

8. ♠ Create a program named in folder `lab3_q8` that keeps track of the items that a wizard can carry.

   Console

```
Wizard Inventory

COMMAND MENU
walk - Walk down the path
show - Show all items
drop - Drop an item
exit - Exit program

Command: walk
While walking down a path, you see a scroll of uncursing.
Do you want to grab it? (y/n): y
You picked up a scroll of uncursing.

Command: walk
```

```
While walking down a path, you see an unknown potion.
Do you want to grab it? (y/n): y
You can't carry any more items. Drop something first.

Command: show
1. a wooden staff
2. a scroll of invisibility
3. a crossbow
4. a scroll of uncursing

Command: drop
Number: 3
You dropped a crossbow.

Command: exit
Bye!
```

Specifications

- Use the file named wizard_all_items.txt that contains a list of all possible items that a wizard can carry. When the user selects the walk command, read these items from the file, randomly pick one, and give the user the option to grab it.
- In another file, store the current items that the wizard is carrying. Make sure to update this file every time the user grabs or drops an item.
- The wizard can only carry four items at a time.
- For the drop command, display an error message if the user enters an invalid number for the item.
- Use functions to organize the code for this program.

9. ♣ Read *Chapter 8: How to test, debug, and deploy a program.*

10. ♢ When you plan the test runs for a program, you should do all but one of the following. Which one is it?
    a. list the valid entries for each test run
    b. list the invalid entries and unexpected user actions for each test run

c. list the expected exceptions for each test run
d. list the expected results for each test run

11. ♢ What line number of the following code contains an error and what type of error is it?

```
1.  double sales_tax(amt) {
2.      double sale = amt + (amt * .06);
3.      return amt;
4.  }
5.
6.  int main() {
7.      cout << "Welcome to the 6% tax calculator!\n";
8.      double total;
9.      cout << "Please enter the total amount: ";
10.     cin >> total;
11.     cout << "The total amount after tax is: " << sales_tax(total);
12. }
```

a. line 1, runtime error
b. line 1, syntax error
c. line 3, syntax error
d. line 11, logic error