

Programming Assignment #4

Hotel Occupancy Report

CS 1428.003, Fall 2019

Instructor: Jill Seaman

Due: before class **Wednesday, 10/16/2019** (upload electronic copy by 9:30am)

Problem:

Your friend works in a hotel lobby, and the hotel needs to be able to run occupancy reports at various times of the day and night. Write a C++ program that will calculate the desired occupancy rates and print a report to a file.

Input: The user should be prompted to input the number of floors the hotel has. Then, for each floor, the program should ask the user for the number of rooms on the floor and how many of them are occupied. **Input validation:** the number of floors must be greater than or equal to 1, the number of rooms on each floor must not be a negative number, and the number of rooms occupied cannot be greater than the number of rooms on that floor. For invalid data, ask the user to re-enter until a valid value is entered.

Processing: Compute the percentage of rooms that are occupied for each floor and for the entire hotel. The percentages should be calculated by dividing the number of rooms occupied by the number of rooms and multiplying by 100. If the number of rooms is 0, then the percentage is 0.

Output: The program should output an occupancy report to a file named **"occupancy_report.txt"**. The first row of the report should be the two column headers "Floor" and "% Occupied". This should be followed by a row for each floor that lists the floor number and the occupancy rate (as a percentage) for that floor. This should be followed by the message **"**low occupancy**"** if the percentage is less than 20. The numerical data in each column should line up under its column header and the percentages should be formatted to have one digit after the decimal. There should be a summary comment after the table indicating the number of total rooms, total rooms occupied, and the percentage of the rooms that are occupied. There must be at least one blank line between the table and the summary comment.

Here is the console output from a sample run of the program:

```
Please enter the number of floors: 4
Please enter the number of rooms on floor 1: 18
Please enter the number of occupied rooms on that floor: 10
Please enter the number of rooms on floor 2: 28
```

```
Please enter the number of occupied rooms on that floor: 20
Please enter the number of rooms on floor 3: 28
Please enter the number of occupied rooms on that floor: 3
Please enter the number of rooms on floor 4: 22
Please enter the number of occupied rooms on that floor: 12
The report is stored in the file occupancy_report.txt
```

Here are the contents of occupancy_report.txt for that execution of the program:

```
Floor    % Occupied
  1      55.6
  2      71.4
  3      10.7  **low occupancy**
  4      54.5
```

```
The hotel has 96 total rooms.
45 rooms are occupied.
46.9% of the rooms are occupied.
```

Here is the console output from a second sample run of the program, to demonstrate the input validation:

```
Please enter the number of floors: 0
The hotel must have at least one floor, please re-enter: 2
Please enter the number of rooms on floor 1: -1
The number of rooms cannot be negative, please re-enter: 3
Please enter the number of occupied rooms on that floor: 1
Please enter the number of rooms on floor 2: 4
Please enter the number of occupied rooms on that floor: 5
The number of occupied rooms cannot be greater than the number of
rooms, please re-enter: 8
The number of occupied rooms cannot be greater than the number of
rooms, please re-enter: 2
The report is stored in the file occupancy_report.txt
```

Here is the console output from a third sample run of the program, to demonstrate a floor with no rooms:

```
Please enter the number of floors: 2
Please enter the number of rooms on floor 1: 0
Please enter the number of occupied rooms on that floor: 0
Please enter the number of rooms on floor 2: 4
Please enter the number of occupied rooms on that floor: 3
The report is stored in the file occupancy_report.txt
```

```
occupancy_report.txt
Floor    % Occupied
  1      0.0  **low occupancy**
  2      75.0
```

The hotel has 4 total rooms.
3 rooms are occupied.
75.0% of the rooms are occupied.

Additional Requirements:

- Your program **must compile** and run, otherwise you will receive a 0.
- Use integer variables whenever the data stored in the variable will not have fractional amounts (only use float or double when necessary).
- Hint: set up your output file and output the column headers BEFORE the loop that processes the floors.
- Hint: output the data to the file immediately after it is computed by your program.
- Note that the column headers may be output as one string literal with spaces embedded in the proper places.
- For partial credit, implement some subset of the features completely. This will probably lead to a better score than implementing every feature poorly.
- Do not use arrays!!

Logistics:

Name your file **assign4_XXXXX.cpp** where XXXXX is your TX State NetID (your txstate.edu email id). The file name should look something like this: assign4_js236.cpp

There are two steps to the turn-in process:

1. Submit an **electronic copy** using the Assignments tool on the TRACS website for this class (tracs.txstate.edu). Submit the .cpp file only.
2. Submit a **printout** of the .cpp file (only!) at the beginning of class on the day the assignment is due. Please print your name on the front page. Print the .cpp file only. Do **not** submit a printout of the output.

See the assignment turn-in policy on the course website (cs.txstate.edu/~js236/cs1428) for more details, including deadlines, penalties, and where to submit printouts after class. **Note: the electronic file cannot be submitted late this time!!**