Donna Graves **COMP 1200 Assignment #1** **Winter 2018**

Due Date: Thurs Feb 15th (before test)

Note: we have been building this assignment using lab exercises 1-3. Carefully, read the program specifications below as the lab exercises may not be exactly as requested for the assignment.

A private hospital would like you to write a program to prepare a report on patient charges. The program should prompt the user for the patient name, room type ('P' for private, 'S' for semi private, or 'W' for ward) and the number of days. The program also should prompt for whether the patient will require a television in the room. The program should continue accepting patient information until they indicate that there are no more patients.

The daily room rates are as follows: A private room is $1000 per day; a semi private room is $750 per day; a ward is $500 per day. A television costs $25 extra each day. A television is <u>not</u> available in a ward. Calculate the patient charge by multiplying the room rate (adjusted for a television, if necessary) by the number of days.

Write each patient name, room type, # days and charge to a file. The output produced should look similar to the following:

```
                 CPA Private Hospital Room Charge
                 *************************
Patient Name    Room Type        # of Days   Room Charge

J. Doe          Private                  5       5000.00
P. Smith        Ward                     1        100.00

Total charges                                    5100.00
```

<u>Code the following functions:</u>

- Write the **main** function to ask for the patient name and if a television is required. Call each of the following functions to complete the program. Use **a do while loop** to handle each new patient. Open a **charges.dat** file for output and print each patient information to the file as shown above.
- Code a function called **getValidRoomType** that will ask the user to enter the room type. Use a while loop to validate to ensure only P, S or W have been entered. Use toupper(). The function does not receive any parameters. It returns a valid room type.
- Code a function called **determineRoomRate** that uses a **switch** statement to determine the room rate. Pass to the function the room type and return the daily room rate.
- Code a function called **getValidDays** that will ask the user to input the number of days the patient will stay. Use a while loop to validate to ensure it is within a valid range. Make up a minimum and maximum value. Ensure your program handles an input failure. You do not have to handle control Z. The function does not receive any parameters. It returns a valid number of days.
- Code a function called **validateReply** that validates the response to the question if the user wants a television or if there are more patients. Use a while loop to validate. Use toupper(). The function <u>receives</u> the reply and returns a validated reply.
- Code a function called **computeCharge** that will compute the patient's room charge. Pass to the function only those values necessary to compute the charge. Return the room charge.
- Code a function called **accumulateCharges** that will accumulate all patient charges. Pass to the function only those values necessary to compute the charge. Return the total charges.

Programming Guidelines:

- **Review the programming standards for the course on Blackboard.** *Not following standards usually results in the most points lost on assignments!*
- Include your name and a short narrative as a comment at the beginning of your program stating what the program does, not how it does it. No title page is required. No handwriting is permitted on submitted programs
- Ensure variable and function names are meaningful. Lowercase or camel-caps. Do not use made up short forms for any part of a name. Do not code any global variables.
- Ensure your program is user friendly.
- **Code all function prototypes even if not required.**
- Code each of the functions outlined above, plus main(). You can add more functions, but you will be marked on following the specifications stated for each function.
- Are brace brackets aligned? The opening brace bracket should be in the same column as the closing bracket. Did you indent inside the brace bracket? *Use Ctrl A + Ctrl K + Ctrl F to ensure alignment is correct.*
- Does the output look similar to the above? Use setw() to print all columns and control the number of decimal places. Ensure you use left/right to justify data in columns as shown. Include the blank lines to separate the data as well.
- Use an if statement to ensure the **charges.dat** file is opened successfully. Crash the program, if it is not opened along with a user friendly message.
- Use C++ shorthand notation when counting (++) and accumulating (+=).
- You can choose any format for the input/output dialogue. It should be unique to your program.
- Add **`system("type charges.dat");`** before your **`system("pause");`** statement to see your output file.
- Use white space to separate related groups of code and functions for improved readability.
- If you are having problems with the assignment, please see me or email me for help prior to the due date.

Note: If your assignment is not working by the due date, hand in what you have completed to earn part marks. Late assignments must be handed in, but are worth zero points.

Hand in: A printout of the program statements, input/output dialogue and output file. Also place a copy in your named folder for the course and copy into my 'dropbox for COMP 1100' folder found under DropboxW7 on the college's network shares.

Do not submit the entire project!

All code is to be free of syntax errors and must be your own work.
Grading is based on well designed, easy to read solutions.
Check out the marking rubric posted in the Assignment folder in Blackboard

"Just because it works does not mean that it is a good solution"