

CS161 - Programming Assignment - TimeFun

The purpose of this assignment is to give you more practice with functions, including using pass-by-reference parameters to get more than one piece of data out of a function.

Overview

Write a program that convert times expressed in 24-hour format to 12-hour format using a.m. and p.m. The program should prompt the user for a time in 24-hour format, validate the input, and then output the equivalent time in 12-hour format. For example, if the input time was 1:34, the output time would be 1:34 a.m. If the input time was 18:00, the output time would be 6:00 p.m. The user should be able to repeat this as often as he or she wishes.

As with all programming projects, you should divide this project into manageable parts and do them one at a time. The pseudocode for this would be this:

```
do
    get time in 24 hour format
    convert time to 12 hour format
    Output time in 12 hour format
    Output time in 24 hour format
while user wants to continue
```

Now that we can write functions, we'll have a separate, well-named function for each of these activities, and main will basically be a loop with a bunch of function calls. And these function calls will be so well named that just reading the code will explain what it's doing and very little documentation (comments) will be necessary.

Note: info on 24 hour time can be found here: http://en.wikipedia.org/wiki/24-hour_clock

Details

You'll need to write the following functions:

getTime24 - Gets valid time (hour and minute) expressed as per 24 hour clock.

This function should prompt the user for a time in 24 hour format, validate the input, and pass back to the caller (by reference) the hour and minute values entered.

Times will be input as two numbers separated by a colon, such as 13:33. To read this input, use a char variable to input the colon. You will not need the colon, but the a variable is still needed. For example:

```
int hour, minute;
char ch;
cout << "Enter a time int 24 hour format (for example 13:45): ";
cin >> hour >> ch >> minutes;
```

If the hour or minute entered are invalid, produce an appropriate error message and re-prompt for valid data. Continue this until valid data is entered, then return.

convertTime24to12: - Converts 24-hour clock hour to 12-hour clock hour

The data into this function is the integer representing the hours in 24-hour format. (we don't need to send the minutes to this function as they don't change) The data out of the function consists of the hour in 12-hour format and the a.m. or p.m. stored as a char, 'a' for a.m. or 'p' for p.m. This function should not output the results to the screen, instead it should send the results back to the caller (by value or call-by-reference; choose the best option)

printTime24 - prints the given time in 24 hour format

Time (hour, minute) should be received as parameters and printed to cout (using streaming I/O) in the following format:

17:56

printTime12 - prints the given time in 12 hour format

Time (hour, minute, am-pm-code) should be received as parameters and printed to cout (using streaming I/O) in the following format:

5:56 p.m.

userWantsToContinue - Determines if user wants to continue.

Prompts the user if he wants to continue. Returns true if user says yes, false otherwise

Sample Output

```
Enter a time in 24 hour format (for example 13:45): 15:30
That time in 12 hour format is: 3:30 pm
That time in 24 hour format is: 15:30
```

Would you like to continue (Y/N)? Y

Enter a time in 24 hour format (for example 13:45): 0:55

That time in 12 hour format is: 12:55 am

That time in 24 hour format is: 0:55

Would you like to continue (Y/N)? N

Extra credit

This extra credit portion should not be undertaken until the program works 100% correctly 100% of the time, and adheres completely to the class coding standards. No extra credit will be given if that's not the case. So test thoroughly!!!! And then test some more. Test all cases.

Now, on to the extra credit....

Modify the time printing functions so the parameters are optional (by providing default values for each of the parameters). Then, if those default values (make them something special like all 9's or something) are received, the function should print the current time instead (google C++ get current time), along with a note saying this is the current time, like this:

The current time is: 3:57 pm.

The current time is 15:57.

Then add calls to these functions (without arguments) to the beginning of main so it prints the current time in both 12 and 24 hour time when it starts.