

CS 135

Programming Assignment 1 (PA1-09/10)

As specified in your syllabus, you must turn your assignment in by 6:00 pm on the due date specified. If it is turned in late, but prior to 12:00 midnight the day it is due, credit will be reduced by 50% of the earned score. Any laboratories turned in more than 6 hours late will not earn any credit.

Objectives:

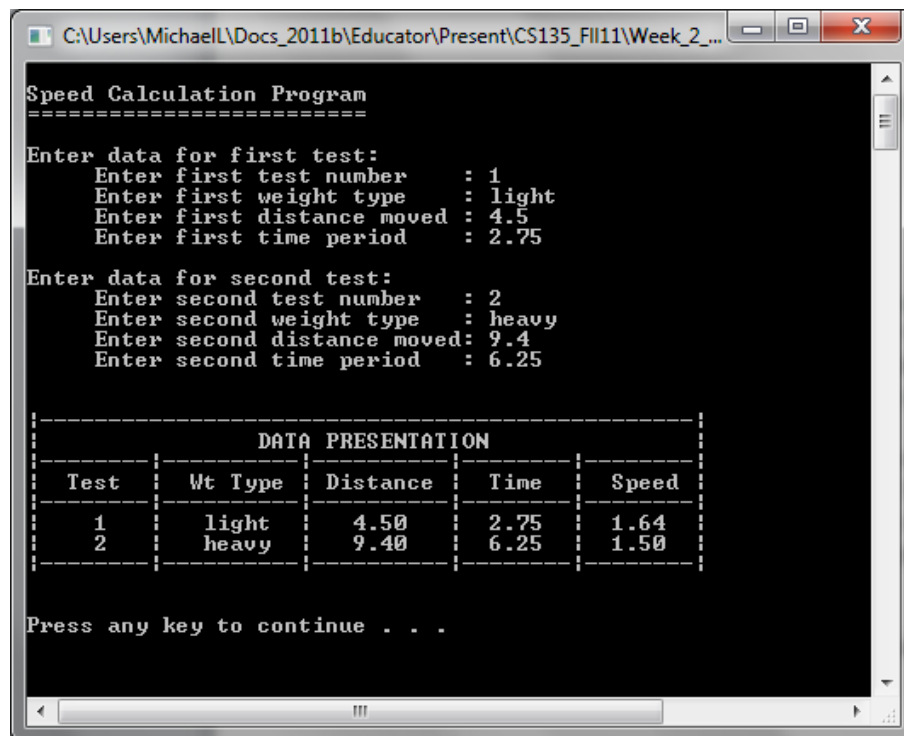
- 1) You will implement I/O operations using the standard input and output functions `cin` and `cout`
- 2) You will use a set of standardized functions to implement console and command-line I/O operations in two different systems
- 3) You will create input and output operations with program abstraction through the use of utility functions
- 4) You will initialize and use variables with a variety of data types
- 5) You will use functions with varying parameters, acquire data from the functions, and assign the results to appropriate variables
- 6) You will conduct simple mathematics in a program
- 7) You will become familiar with the use of a standard explanatory commenting process for developing functions
- 8) You will continue to use a common IDE to develop and test programs

Tasks:

Using cin/cout and standard console I/O

- 1) First, remember to create a new folder under your CS135 folder and name it "Week 2". You will need this to get started. Your program name will be `speedcalc.cpp`.
- 2) Your first program is called `speedcalc.cpp` and takes input related to Physics problems and outputs them in a well organized way, along with a calculated result. The math is easy to do in this program, but most of your work will be using functions to acquire input and provide a nice looking display.

- 3) Download the `formatted_cmdline_io_v07.h` file from WebCampus. You may need to copy the text and paste it into your Dev C++ IDE, but if you do, make sure you save the file as a "Header file" as instructed in the PA0 assignment. This file has a number of utility functions that will help you implement many input and output operations.
- 4) There are instructions in this file that support the use of the available functions and there are function descriptions for each one. You will be using function descriptions like these later on when you create your own functions, so it is worth your while to become familiar with them. Reading the function specifications also helps you understand what each function is used for and how you can use it.
- 5) Your assignment this week will be to create a program that prompts for a set of data, and displays it in the format shown below. Your program must meet the following specifications so you should read through all of these instructions before you begin.



```
C:\Users\MichaelL\Docs_2011b\Educator\Present\CS135_Fll11\Week_2_...
Speed Calculation Program
=====
Enter data for first test:
Enter first test number      : 1
Enter first weight type      : light
Enter first distance moved   : 4.5
Enter first time period      : 2.75

Enter data for second test:
Enter second test number     : 2
Enter second weight type     : heavy
Enter second distance moved  : 9.4
Enter second time period     : 6.25

-----
DATA PRESENTATION
-----
| Test | Wt Type | Distance | Time | Speed |
|-----|-----|-----|-----|-----|
| 1     | light   | 4.50     | 2.75 | 1.64  |
| 2     | heavy   | 9.40     | 6.25 | 1.50  |
|-----|-----|-----|-----|-----|

Press any key to continue . . .
```

- 6) The program design will be provided to you this week in the form of the Six Step Programming Process' steps 1, 2, and 3. This program is very simple, but you will see how a program should be organized using modular structure, and how the appropriate modules (i.e., functions) can be used to solve the problem. Your Programming Assignment (PA) for this laboratory will be the step 4 of the Six Step Programming Process. Note that next week, you will be responsible for both the DA and the PA. The following instructions are written as if you would be designing the program yourself. So although the design is created for you, spend some time looking this over as this will be the form of all your DAs and PAs throughout the semester
- 7) Your program design must include the following.
 - a) For each "first" item, you must use the standard `cout` and `cin` operation as specified below to prompt the user and acquire input:
 - i) for "Enter first test number:", use `cout` and `cin` to acquire an integer value
 - ii) for "Enter first weight type:", use `cout` and `cin` to acquire a string value
 - iii) for "Enter first distance moved:", use `cout` and `cin` to acquire a floating point (double) value
 - iv) for "Enter first time period:", use `cout` and `cin` to acquire a floating point (double) value
 - b) For each "second" item you must use the "promptFor..." functions from the formatted command line header file to prompt and acquire data from the user:
 - i) for the "second test number", use a "promptFor..." function to acquire an integer value (e.g., "`promptForInt`")
 - ii) for the "second weight type", use a "promptFor..." function to acquire a string value
 - iii) for the "second distance moved", use a "promptFor..." function to acquire a floating point value
 - iv) for the "second time", use a "promptFor..." function to acquire a floating point value
- 8) Once you have acquired all this data and assigned it to appropriate variables, you must calculate the speeds of the two conditions. The speed calculation for each item is distance over time.
- 9) For the output display, use only the "print..." functions (e.g., "`printInt`") to output all of the data items in the table format shown. All displayed data is centered horizontally within the boxes. These functions will make the process very easy, but you will need to practice with them to become comfortable. Note that you will be using these or comparable functions throughout the semester; this is a powerful opportunity to learn about program abstraction and organization within an easy program.
- 10) Make sure that your input and output processes look as much like the example above as possible; in most cases, it should look exactly the same. Responses to prompts **MUST** be on the same line as the prompt (as shown), and the table output must be well organized.

- 11) Important Note: You MUST use the Design Assignment as provided; you may not make any design changes to it. This will not be difficult as the DA provides almost line-by-line specifications for what program code is needed in a particular place. That said, if your program code does not align with the given DA, your credit will be reduced.
- 12) Create three example screenshots of this program using the data above for your first, and your own data for the next two. Remember to annotate each screenshot.
- 13) Note that the Rubric for each of the assignments is always posted along with the assignments. It is very worth your time to review in advance the way that your assignment will be graded.

Turning in your Programming Assignment:

Information:

Week: 2

Laboratory: Week 2

Programming Assignment: 1

Due Date: 09/10, 6:00 pm

To turn in:

1. The Word file containing the following:
 - a. There should be at least three (3) screenshots for the programs as specified above
 - b. Remember to clearly annotate every displayed result
2. The executable files:
 - a. speedcalc.exe (speedcalc_s4.exe is acceptable)
3. The source code files:
 - a. speedcalc_s4.cpp

These files must be compressed and uploaded as one zip file. To do this, select all of the required files, right click on them, and select “Send To”, then select “Compressed (zipped) Folder”.

Once the folder is created, it will be placed in the same folder in which you are working.

Change the name of the zipped folder to “LastnameFirstname_PA1” as follows:

“LeveringtonMichael_PA1” (no quotes). After you have renamed the zipped folder, double click on it to verify that it has all the files it is supposed to have.

For information on how to turn in Programming Assignments, refer to the "How to Turn in Programming Assignments" in the "General Course Information" folder