Lab1: Conversion CS-116

The purpose of this lab is for you to familiar with your C++ development environment. It will help to refresh some of the very basic C++ knowledge.

Write a program that do the following unit conversion based on user menu selection: (use while or do-while loop until the user enter 'Quit' selection)

1. Seconds to hours, minutes and seconds (must implement as a function)

   - inputs an integer that represents a length of time in seconds. The program should then output the number of hours, minutes, and seconds that corresponds to that number of seconds.
   - **Test Cases:**
     - If the user **input 50391** total seconds then the program should output 13 hours, 59 minutes, and 51 seconds.
     - If the user input is -80, then your program should ask user to reenter a valid number.
   - **Your program must check for valid input (positive integer only)**

2. Fahrenheit to Celsius **(must implement as a function)**

   - Inputs an Fahrenheit the program will convert to Celsius.
   - For example, input 82 the program should output 27.8
   - **°F to °C – Deduct 32, then multiply by 5, then divide by 9**
   - $T_{(°C)}$ = (82°F - 32) × 5/9 = 27.8 °C
   - Your program must check for valid input (number only)
   - Hint: use *cin.fail()* to check for valid input based on the cin data type (cin.clear(), cin.ignore() and cin.fail())
   - **Test cases:**
     - If user enters non-digits such as "abc" then your program should ask user to enter a valid number.
     - If user enters 82, the program should output 27.8 C

3. Celsius to Fahrenheit **(must implement as a function)**

   - Inputs Celsius the program will convert to Fahrenheit.
   - For example, input 12 the program should output 53.6
   - °C to °F - Multiply by 9, then divide by 5, then add 32
   - Convert 12 degrees Celsius to degrees Fahrenheit:
   - $T_{(°F)}$ = 12°C × 9/5 + 32 = 53.6 °F
   - Your program must check for valid input (number only)
   - **Test cases:**

- If user enters non-digits such as “fifty” then your program should ask user to enter a valid number.
- If user enters 40, the program should output 104.0 F

Modify the printMeFirst() function below to include your personal information, and call the function 1st before anything else. Below is the sample of the program template

```
#include <iostream>
#include <string>
#include <iomanip>
#include <ctime>

using namespace std;
```

You can use the following template for all your function documentation format

```
/**
   Print out the programmer's information such as name, class information
   and date/time the program is run

   @param name -  the name of the programmer
   @param courseInfo -  the name of the course
   @return - none
*/

void printMeFirst(string name, string courseInfo)
{
   cout <<" Program written by: "<< name << endl; // put your name here
   cout <<" Course info: "<< courseInfo << endl;
   time_t now = time(0);  // current date/time based on current system
   char* dt = ctime(&now); // convert now to string for
   cout << " Date: " << dt << endl;
}

main()
{
        // pass your name, your class info to function so it will print them out
        // always call this function in all of your lab

        printMeFirst("Your_FirstName  Your_LastName","Lab1: CS-116 - Conversion");

        //  write your codes below
        //

}
```

Use the following as test cases for your program and your program should output similar to the one below:

Sample program output screenshots below:

```
 Program written by: Ron Sha
 Course info: Lab 1: CS-116 - Conversion
 Date: Wed Jan 30 13:25:52 2019

Choose a selection below:
        1 - Second to hours, minutes, and seconds
        2 - Fahrenheit to Celsius
        3 - Celsius to Fahrenheit
        4 - Quit
1
Input seconds (positive integer):
-80
Please enter a positive number.
Input seconds (positive integer):
50391
50391 seconds is: 13 hours, 59 minutes, 51 seconds
```

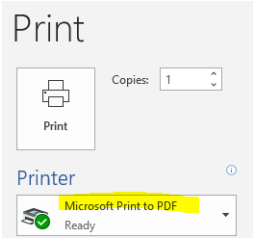
```
Choose a selection below:
        1 - Second to hours, minutes, and seconds
        2 - Fahrenheit to Celsius
        3 - Celsius to Fahrenheit
        4 - Quit
2

Enter a Fahrenheit value: abc

Enter a valid number

Enter a Fahrenheit value: 82
82 F is equal to 27.8 C
```

```
Choose a selection below:
        1 - Second to hours, minutes, and seconds
        2 - Fahrenheit to Celsius
        3 - Celsius to Fahrenheit
        4 - Quit
3
Enter a Celsius value: fifty

Enter a valid number
Enter a Celsius value: 40
40.0 C is equal to 104.0 F

Choose a selection below:
        1 - Second to hours, minutes, and seconds
        2 - Fahrenheit to Celsius
        3 - Celsius to Fahrenheit
        4 - Quit
4
```

Lab Submission: See the lab submission requirements published in canvas.

To submit your assignment in canvas, you must submit TWO files (one pdf and one zip) as follows:

1. **Attach pdf file** which contains source codes you have written and program output screenshot so I can easily read in one file. You can use a word editor to place all the required programs and screenshots, and then use 'Print' to 'Microsoft Print to PDF' to save to a pdf file



The pdf file MUST have the following sections (1. Program Description, 2. Program Source Code and 3. Program Output).

1. Program Description

- brief *description* of the purpose of the *program and*

- an explanation of what your software does and what problem it solves

2. Program Source Code

- Include all the source codes (program files) you have written for this lab (screenshots are ok)
- Your program must have adequate documentation for your source codes:
  - Program description – see above on Program Description
  - **You must put the following function headers for each function (the function header MUST be placed just above the function declaration in your source code). For Functions, it must have:**
    - *Function name: name of this function*
    - *Function description: the purpose of this function and how to use it*
    - *@param param_name and what the parameter/argument is used for*
    - *@return what is returned from the function*
- Make sure your screenshots are readable (not too small)

Below is an example of source code

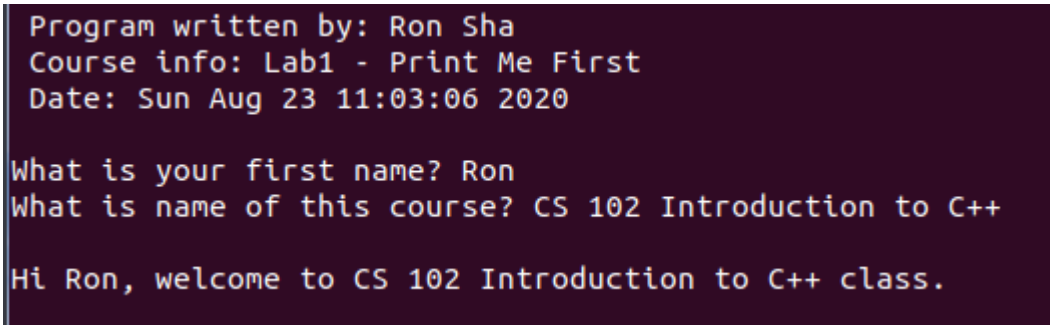
File: print_me_first_main.cpp (list the program file individually)

```
1  //print_me_first_main.cpp
2  #include <iostream>
3  #include <string>
4  #include <iomanip>
5
6  using namespace std;
7
8  /**
9   * @Purpose - this function print out the person who wrote the program,
10  * and date/time the program run.
11  * @parm - name - the author of the program
12  * @parm - courseInfo - the name of the course
13  * @return - none
14  * @author - Ron Sha
15  **/
16
17 void PrintMeFirst(string name, string courseInfo)
18 {
19     cout <<" Program written by: "<< name << endl; // put your name here
20     cout <<" Course info: "<< courseInfo << endl;
21     time_t now = time(0); // current date/time based on current system
22     char* dt = ctime(&now); // convert now to string for
23     cout << " Date: " << dt << endl;
24 }
```

```
25
26  /**
27   * @Purpose - this main program is used to test and ensure your have
28   *     setup your C++ development environment to ensure all is working.
29   *
30   *     It will first call the PrintMeFirst function to prin out
31   *     programmer's name and any relevent information.
32   *
33   *     To test the programming environment is setup correctly, this
34   *     program will also testing out the standard input by asking
35   *     the user to enter your name and course information.
36   *
37   *     After getting user's inputs, then the program will print out
38   *     what user's entered in the standard output.
39   *
40   * @parm - argc - Tells you how many command-line arguments there were.
41   *                It is always at least 1, because the first string in
42   *                argv (argv[0]) is the command used to invoke
43   *                the program.
44   * @parm - argv - argv contains the actual command-line arguments as
45   *                an array of strings, the first of which (as we have
46   *                already discovered) is the program's name.
47   * @return - none
48   * @author - Ron Sha
49   **/
50
51  int main(int argc, char* argv[])
52  {
53      string myName, courseInfo;
54
55      /* You MUST use the PrintMeFirst function in all of your labs
56       * in this class
57       */
58      PrintMeFirst("Ron Sha", "Lab1 - Print Me First");
59
60      // Below is where you place your programs for this lab
61      cout << "What is your first name? ";
62      cin >> myName; // cin is used gets a word, space is separator
63      cin.clear(); // clear error flag
64      cin.ignore(1024, '\n'); // ignores remainder of stream input
65
66      cout << "What is name of this course? ";
67      getline(cin, courseInfo); //getline is used to get entire line input
68      cout << endl;
69      cout <<"Hi " << myName << ", welcome to "
70           << courseInfo << " class.\n";
71  }
```

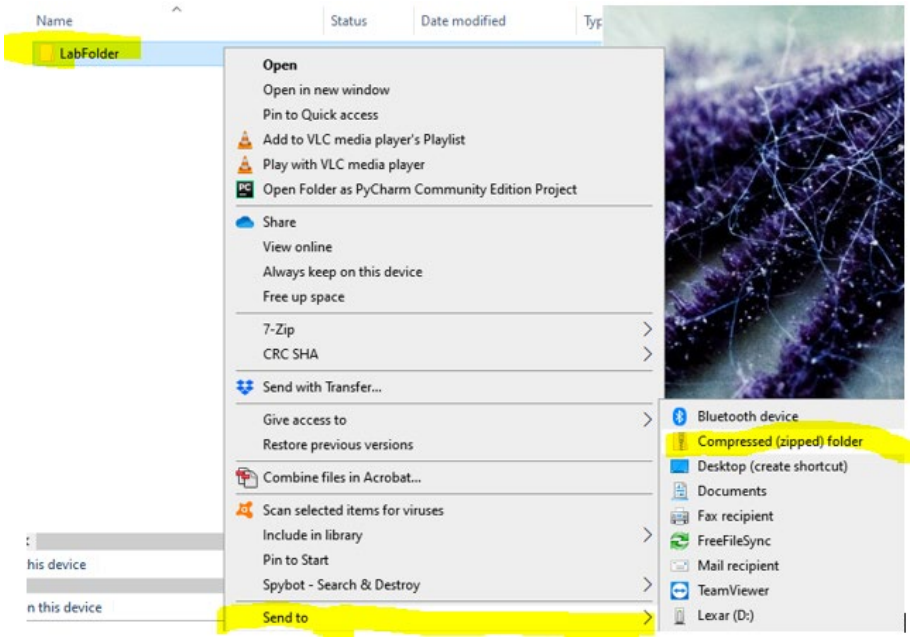
3. Program Output

- Attach all the program outputs (screenshots)
- Don't place Source code and program output side by side as it is not readable in screenshot
- Make sure your screenshots are readable (not too small)
- Your main program outputs MUST include your name printout (use the print_me_first function/program).
- Below is an example of program outputs:



2. **Attach zip file** which contains all your source code (you can zip the folder) and functions. Even if you only have one source file, you MUST still do a zip file of the folder. I must be able to compile and run your program from all the source code programs after I unzip your zip file.

   You should create a folder for each lab, and place all your programs, functions and all other files related to this lab in this lab folder. To submit the lab folder, you can use "Send to -> compress" in window file explorer to create a zip file of the folder.



3. **Now you can upload both the pdf file and zip file separately as your assignment submission in canvas.**