CS 150 **Lab Assignment 1** **Spring 2016**

In lab this week, you will work individually. We will study and practice working with source code and running c++ programs in our IDE, using variables, named constants, and assignment statements, basic I/O operations (cout , cin), and simple arithmetic expressions.

- **Lab 1.1** **W**riting programs, Solving problems, Algorithm Design - all simple. (conversion program)
- **Lab 1.2** **E**xploring the basic elements of a C++ program. statements, endl, cin, cout, escape sequences.
- **Lab 1.3** **I**dentifying Data Types, and Reserved Words.
- **Lab 1.4** **A**rithmetic operations, operator precedence, type casting, and the string data type.
- **Lab 1.5** **R**unning and examining C++ programs. Pre-processor directives. Good program style.

Lab 1.1 Instructions

Write a program that takes as input a given length expressed in feet and inches, and converts and then outputs the length in centimeters. The user input is length in feet and inches. The program computes the equivalent length in centimeters, using the **conversion formula**: ***One inch is equal to 2.54 centimeters***. The program displays this calculated value as the output. The algorithm used for the calculation is shown below, followed by the overall algorithm used for the program solution. You will see these steps in the source code template provided for the lab. Read through them, so that you are familiar with the problem solving steps required for the solution.

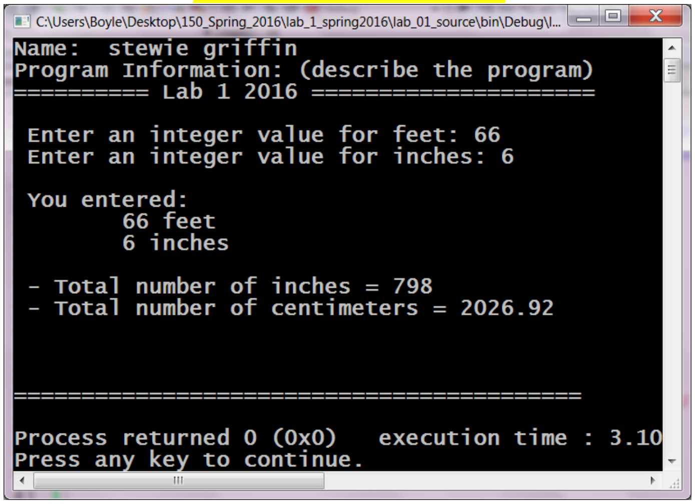
The *algorithm* to calculate the equivalent length in centimeters follows:

1. Get the length in feet and inches.
2. Convert the length into total inches (Multiply the number of feet by 12, and add given inches)
3. Convert total inches into centimeters (Using the conversion formula)
4. Output centimeters

The overall algorithm for the program solution:

1. Prompt user for input
2. Get data
3. Echo the input (output the input)
4. Find length in inches
5. Output length in inches
6. Convert length to centimeters
7. Output length in centimeters

Example Program Output



C:\Users\Boyle\Desktop\150_Spring_2016\lab_1_spring2016\lab_01_source\bin\Debug\l...

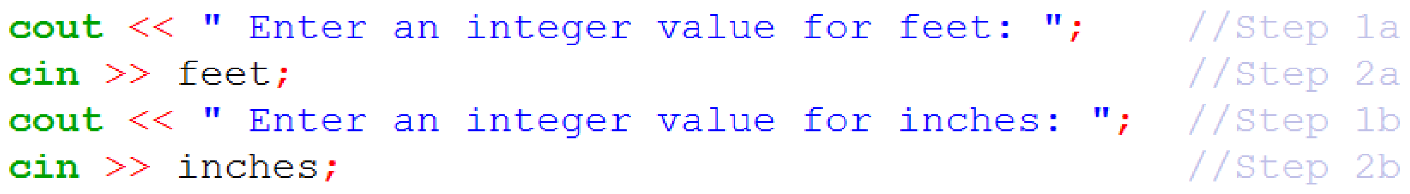
```
Name:  stewie griffin
Program Information: (describe the program)
========== Lab 1 2016 ======================

 Enter an integer value for feet: 66
 Enter an integer value for inches: 6

 You entered:
        66 feet
        6 inches

 - Total number of inches = 798
 - Total number of centimeters = 2026.92



==========================================

Process returned 0 (0x0)   execution time : 3.10
Press any key to continue.
```

1. Read this lab document completely before you begin working on the source code program which we have provided – aka "the template". Open the source code template and save a copy as lab01_2016_F_LName.cpp, replacing the F with your first initial, and LName with your last name. That is the required naming convention for our labs. You will lose points if your file is not named correctly.

2. Read the template, and specifically look for comments which contain key information to complete the lab and solve the problem. Refer to the sample output above.
3. Fill in the preliminary comments at the top of the file, including your name, and program description. This is a critical step that personalizes your professional solution. You will lose points if it is not complete.

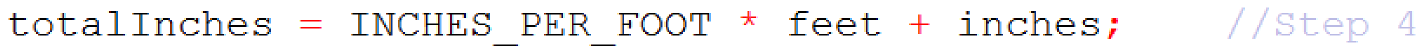
Locate and complete the missing parts of the program, using the code snippets provided below.

Steps 1 and 2 – Fill in the code to complete the missing steps 2a and 2b. This is using the **cin** object and input stream to capture user input and store it in our variables called feet, and inches.



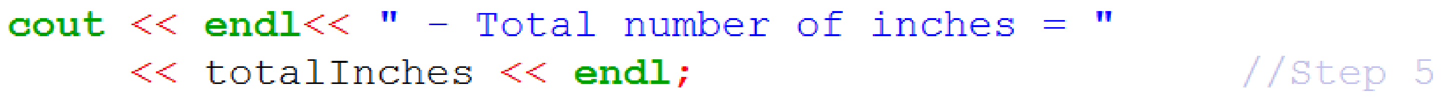
```
cout << " Enter an integer value for feet: ";      //Step 1a
cin >> feet;                                        //Step 2a
cout << " Enter an integer value for inches: ";    //Step 1b
cin >> inches;                                      //Step 2b
```

Step 4 - Fill in the code to calculate the total inches.



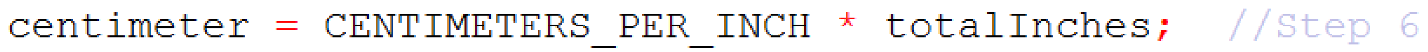
```
totalInches = INCHES_PER_FOOT * feet + inches;     //Step 4
```

Step 5 - Complete the code to output the total number of inches calculated.



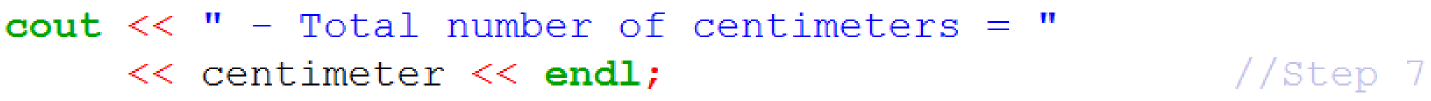
```
cout << endl<< " - Total number of inches = "
     << totalInches << endl;                        //Step 5
```

Step 6 – Complete the code the code to calculate the total centimeters.



```
centimeter = CENTIMETERS_PER_INCH * totalInches;   //Step 6
```

Step 7 - Fill in the code to output the results.



```
cout << " - Total number of centimeters = "
     << centimeter << endl;                         //Step 7
```

Fill in all of the steps above, and compare your program output to the sample output provided on page one of this lab. Adjust your source code if needed, and as necessary to duplicate the required output.

Perform the following and observe what happens. Your TA can help step you with these steps.

- Press F2 (and then again) You should see the message window appear.
- Press shift-F2 (and then again) You should see the project management window appear.
- Put your cursor on a source code line, and then hit cntrl-D. The line should duplicate.
- Remove a semi-colon from the end of one statement and observe the error in the message window, when you attempt to build and run the code. Induce and repair errors in the source code to observe the messages output and the indications of failure provided in the message window.
- Remember to save your source code solution with the proper name before submitting (described above).

Your TA will instruct you on how to submit your program solution.

The following parts of the lab do not have to be submitted for credit. They are provided as tools for you to study the course materials. (and who knows what may show up on an exam!)

Lab 1.2 In this part of the lab, we will become familiar with the 'tokens' (reserved words, special symbols, and identifiers) used in a C++ program. Indicate whether the following representations is a reserved word (Appendix A), special symbol, or an identifier, or is invalid. Use chapter 2 as a reference, along with Appendix A.

TOKEN	Special Symbol	Reserved Word Symbol	Valid Identifier	Invalid
1. SALARY				
2. cat!				
3. float				
4. void				
5. r2d2				
6. 2hot4u				
7. %				
8. 7days				
9. @home				
10. s_u_m				
11. pattern				
12. +				
13. Do				
14. <=				
15. false				

Lab 1.3 In this part of the lab, we review the simple data types found in a c++ program.

Value	Boolean	Integral	Floating Point	Character	Invalid
1. 3					
2. -6987					
3. 4.51					
4. false					
5. 'F'					
6. 9.4E4					
7. 3,345					
8. tralse					
9. 'u'					
10. 451					
11. 0.0					
12. true					
13. '+'					
14. 5.667					
15. 'off '					

16. The implicit conversion of one data type to another data value is called _______________________.
17. The explicit conversion of one data type to another data value is called _______________________.
18. The following is an example of the string data type: **"77"** True False
19. The following is an example of the string data type: **false** True False
20. The following is an example of the string data type: **"Hello ODU"** True False

Lab 1.4 In this part of the lab, we review the arithmetic operations, operator precedence, type casting, and the string data type concepts covered in chapter 2. Indicate the result of the following expressions.

	Expression	Result
1	20 % (4 - 2)	
2	20 / 4.0 * 6.4 /2	
3	4 % 5	
4	3.0 * (6 /24)	
5	3 - (3 + 3.0) * 10/3	
6	12 % 4	
7	7 * 6 / 21 /3.0	
8	11 % 4 * 3.0	
9	17 % 4 /3	
10	5 % 6 /4	
11	static_cast<int> (3.7)+5.3	
12	static_cast<int> (3.7 + 5.3)	
13	static_cast<double> (5/2)	
14	static_cast<char> (65)	
15	static_cast<float> (4)/3	

Lab 1.5 In this part of the lab, we study source code examples in detail, and compile and run them in our IDE.

1. Source code files for this lab are included in the zip file:
    a. **Create a CodeBlocks project** called Lab01 (on your Z drive on the CS network).
    b. Open each of the example source code files in CodeBlocks.
    c. Examine the source code of the programs line-by-line. Read all of the comments. Instructions for completing parts of the labs may be contained in the source code and comments of the examples.
2. Compile and run the programs. In the table below, provide a brief description of what each of the example programs is demonstrating. (use your textbook, and the source code comments to help).

• **Example2_1**	
• **Example2_3**	
• **Example2_4**	
• **Example2_4_Mod**	
• **Example2_5**	
• **Example2_8**	
• **Example2_9**	
• **Example2_13**	
• **Example2_17**	
• **Example2_18**	
• **Example2_19**	
• **Example2_22**	
• **Example2_26**	
• **Example2_29**	
• **Example2_44**	
• **Example2_45**	
• **Example2_Ex5**	
• **Example2_MakeChange**	
• **Example2_Convert**	