

CS1580 Lab 2

1 Lab Topics

1. C++ variables
2. Taking input
3. Arithmetic operations
4. Debugging and error fixing
5. Practice good coding standards and follow the CS1570 Coding standard.
6. Assignment submission in git and canvas.

2 Restaurant rating calculator

Suppose the rating of a restaurant is computed by following formula:

$$\text{Number of visits per week} + \text{number of social media tags per week} \\ + \text{number of items in the menu} \times \text{average price of an item}$$

Here **average price of an item** is a decimal number, and the rest of the variables are integers. You have to write a program that takes all four values as user inputs and prints the final rating.

Once your code compiles and runs, try to break it by

1. Removing a semicolon
2. Splitting a statement into two lines (Be sure to split a statement that causes compilation error).

Take screenshot of the first few lines of errors for both cases.

**** Remember to fix those errors before submission. The purpose of breaking the code is to take the screenshots and upload those screenshots in canvas. But your final code should not have any errors.**

3 Conversion Calculator

Write a program that calculates and displays various conversions and calculations based on user input for a given distance in miles:

1. Convert the distance to kilometers.
2. Calculate the time it takes for a car traveling at a constant speed of 60 miles per hour to cover that distance in hours.
3. Determine the equivalent number of football fields (1 football field = 100 yards) the distance represents.
4. Find the area of a square with a side length equal to the converted distance in kilometers.
5. Calculate the circumference of a circle with a radius equal to half of the converted distance in kilometers.

Example Output:

Please enter the distance in miles: 100

1. Distance in kilometers: 160.934 kilometers
2. Time to cover the distance at 60 miles per hour: 1.667 hours
3. Equivalent number of football fields: 18.52 football fields
4. Area of a square with side length equal to the converted distance: 25916.55 square kilometers
5. Circumference of a circle with a radius equal to half the converted distance: 80.467 kilometers

4 How to get full marks

To get a 100% on this lab your code should:

1. Use good variable names such that one can easily understand a variable's purpose just by looking at the name
2. The program needs to be intuitive (e.g., display proper messages while you are taking user input or printing the result)
3. Follow all good coding conventions such as proper indentation.
4. Comment your code properly (do not write comments for things that are obvious)
5. Push your most recent code in git and submit through canvas as well. The canvas submission should include following files:
 - a. The cpp file downloaded from git
 - b. 3 image files (one of them is the actual output and other two are error screenshots)

5 Lab Preparation

First, this starts by making a folder for this lab. Only material for this week's lab should be placed in this folder. For example, if you made a cs1580 folder last week, and cloned the repository that I created for you, then you can skip step 1-4 and start from step 5

1. `cd SDRIVE`
2. `mkdir cs1580`
3. `cd cs1580`
4. clone the repository repository named **2021-SP-YOUR_SECTION-Labs-YOUR_USER_ID**. If cloned correctly, you will see a folder named 2021-SP-YOUR_SECTION-Labs-YOUR_USER_ID being created. Then `cd` to that folder.
5. `mkdir lab2`
6. `cd lab2`

6 Material review for today's lab

1) cin and cout

cin and cout are special variables in C++ that takes input from the user and display information to the user, respectively. They are *streams*. The << and >> show how data flows into or out from these streams.

- Data comes in from a cin and the >> characters should always be used with cin.
- Data goes out to the screen with a cout and the << characters should always be used with a cout.

2) Coding standard

The coding standard is an 11-page document listed on Professor Price's website (http://web.mst.edu/~cpp/cpp_coding_standard_v1_1.pdf). Eventually, you'll be expected to follow the entire coding standard. For now, however, make sure your lab meets the following requirements:

- ***The comment section at the top of your lab is filled out and all the appropriate sections are listed (see below).***
- ***Curly braces { and } go on their own lines.***
- ***You have to indent your code by double spaces after a curly brace***
- ***The name used for variables in the sample program makes sense.***
- ***Do not type pass 80th column***
- ***Comment anything that is not obvious or self-explanatory***

3) Indentation and curly braces

Anything between a pair of curly braces { and } should be indented **2 spaces**. Each time you open a curly brace, indent over two more spaces. Do not use tabs—these print differently than they are viewed, and will result in line wraps!

Additionally, curly braces should always go on their own line.

```
int main()
{
    // I'm inside some curly braces, so I've indented two spaces!
    cout<<"I'm the coolest kid on my block!"<<endl;
    return 0;
}
```

7 How to submit

- **Submit in git:**
 - git add .
 - git commit -m "your message"
 - git push

- It will ask you for your username and password
 - While you type in password, it does not show it and that is completely normal!
- **Submit in canvas:**
 - Take a screenshot of the output
 - Upload your most recent code files (by downloading it from git-classes website) as well as output screenshot on canvas