LAB DIRECTIONS for LAB 2/26/2015

Based on Lesson Set 5

1. Create a folder named **Lesson Set 5**. Put all files you create for this lab in this folder.

2. Download the **Lesson Set 5 source files** from ilearn and extract them into your Lesson Set 5 folder on your computer or jump drive.

3. Bring in program **`while.cpp`** from the Lesson Set 5 source files folder.
   a. Run the program as it is to notice that the program is not user friendly. Add to the code to make the program more user friendly.

   b. Compile and run the program.

   c. Take a screen capture of the program running and paste in a word document titled **Lesson Set 5 Screen Captures**.

4. Bring in the program **`dowhile.cpp`** from the Lesson Set 5 source files folder.
   a. Fill in the indicated code (directed by comments) to complete the `dowhile.cpp` program.

   b. Compile and run the program.

   c. Replace the line
   `if (validBeverage == true)`
   with the line
   `if (validBeverage)`

   d. Compile and run the program with all valid input and also invalid input. Take a screen capture of the program running and paste in the document titled **Lesson Set 5 Screen Captures**.

5. Bring in program **`for.cpp`** from the Lesson Set 5 source files folder. This program has the user input a number n and then finds the mean of the first n positive integers.

   a. Modify the code to change the following line
   `mean = static_cast<float>(total) / value;`
   with this one
   `mean = total / value;`

   b. Compile and run the program. Take a screen capture of the program running and paste in the document titled **Lesson Set 5 Screen Captures**.

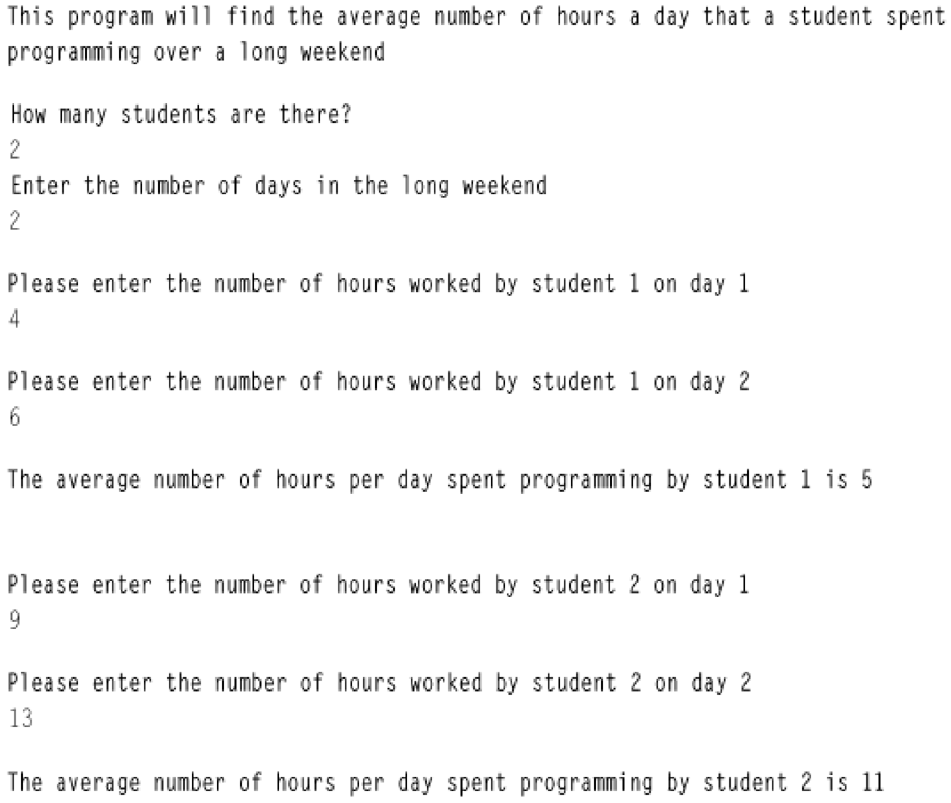
   c. Change the code back to using static_cast to calculate the mean.

   d. Compile and run the program with the same input as in 5b. Take a screen capture of the program running and paste in the document titled **Lesson Set 5 Screen Captures**.

6. Bring in program **`nested.cpp`** from the Lesson Set 5 source files folder.

   a. Note that the inner loop of this program is always executed exactly three times – once for each day of the long weekend. Modify the code so that the inner loop iterates n times, where n is a positive integer input by the user. In other words, let the user decide how many days to consider just as they choose how many students to consider.

Sample Run:



```
This program will find the average number of hours a day that a student spent
programming over a long weekend

 How many students are there?
 2
 Enter the number of days in the long weekend
 2

Please enter the number of hours worked by student 1 on day 1
4

Please enter the number of hours worked by student 1 on day 2
6

The average number of hours per day spent programming by student 1 is 5


Please enter the number of hours worked by student 2 on day 1
9

Please enter the number of hours worked by student 2 on day 2
13

The average number of hours per day spent programming by student 2 is 11
```

b. Modify the program now so that it also finds the average number of hours per day that a given student studies biology as well as programming. For each given student include two prompts, one for each subject. Have the program print out which subject the student, on average, spent the most time on.

c. Compile and run the program. Take a screen capture of the program running and paste in the document titled **Lesson Set 5 Screen Captures**.

7. Bring in **`billfile.cpp`** and **`transaction.txt`** from the Lesson Set 5 source files folder.

   a. Fill in the statements that will read the data from the file and print the following to **`bill.txt`**. Your program will create bill.txt.

   ```
   The total bill is $241.56
   ```

   b. Compile and run the program. Take a screen capture of the program running and paste in the document titled **Lesson Set 5 Screen Captures**.

What to Turn In: (by Wednesday, March 4, 2015)

- **while.cpp**
- **dowhile.cpp**
- **for.cpp**
- **nested.cpp**
- **billfile.cpp**
- **transaction.txt**
- **bill.txt**
- **Lesson Set 5 Screen Captures**