CS 159 – Spring 2022 – Lab #1

Overview of Current Lab:

1. **Important** – submit your attendance at the start of every lab meeting: https://www.cs.purdue.edu/cs159/lab/submit_attendance.php
2. Please read over the collaborative teaming expectations.
3. Work with your team on the written problems found in this document.
4. Begin your work on the programming assignment.
5. Complete the Collaborative Group Communication form and submit to your lab instructor.
6. During the final 10 minutes of your lab today you will complete the first lab quiz.

Collaborative Teaming:

- **Why utilize and rotate roles among the members of a team?**
  - The use of, and rotation through, the roles described will allow each member to gain experience with the entire process of developing a solution to a programming problem. As a team there should be a shared interest in strengthening each individual member such that they are able to continue to contribute as problems become longer and increase in complexity. Individuals should insist on a rotation of roles during the entire development process to better prepare for homework programming assignments which are individual efforts.
  - The roles are designed to provide the opportunity for individual members to learn about the problem-solving process and the implementation of a solution using the tools of the course. The roles do not emphasize efficiency as each lab programming assignment is due approximately one week after it becomes available. Do not allow your desire to complete every assignment quickly be at the expense of learning the material.
- **Groups are expected to communicate to share their ideas when it comes to solving the conceptual and programming problems associated with this lab.** You may find a collaborative document to be a helpful way to share thoughts on the written problems and to formulate the logic for the programming problem.
- **As a group you must determine who will make the final submission for your group**, when that submission will be made, and how the concerns regarding submission will be communicated with the other members.
- **What if a partner does not respond to your communication?** Then the remaining active partners need to be prepared to proceed on the assignment to meet the deadline.

Lab Quiz #1 – 5 points possible: The lab quiz will be made available on Brightspace (Week #2 module) during the final 10 minutes of your lab today. The quiz will emphasize material from chapter 1, chapter 2, and the course programming and documentation standards relevant to the lab programming assignment. **Lab quizzes are individual efforts and you may not use any resources while completing the quiz.** Questions are presented one at a time and cannot be revisited. Be sure to save your answers to each question and to finish your quiz to ensure it is submitted for grading. Most problems on lab quizzes will be multiple-choice or true-false.

- Quizzes are password protected and will be provided by your lab instructor.
- The time at which you take your quiz is monitored and students completing their quiz at an unexpected time are subject to loss of points and a potential academic integrity inquiry.

Lab Programming Assignment #1 – 5 points possible: Every lab programming assignment is due 30 minutes prior to the start of your next lab meeting. Program 2-7 on pages 71-72 of your C programming text is a part of the written problems that follow and may provide a good example from which you can begin to solve this first lab problem.

- **How might collaboration be useful on this particular programming assignment?** This assignment is not particularly long or logically challenging but it will test your understanding of input, formatting output, and applying the course standards.

CS 159 – Collaborative Group Communication

Name	Purdue career account name	Who will make the final submission of this lab assignment?
		The first individual listed here is responsible for the final submission of this lab programming assignment.
		These partners must be responsible for submitting at least one future lab assignment for this team.

Each member should initialize affirming the following:

Every member has the contact information of the other lab partners.			
The collaborative team has been created and finalized in Vocareum.			
How will members of the group be contacted should a concern arise with the assignment? Be specific (text message, e-mail, groupme, slack).			
When and where is the group meeting next?			
When will the final electronic submission be made?			
Who is responsible for bringing the C programming text and class notes to the next lab meeting?			

Solve the following problems related to material found in Chapter 1, Chapter 2, and the course standards.

Statement	True / False
Section 1.1	
The CPU is responsible for executing instructions for the computer.	
Main memory is a place where the programs and data are stored temporarily during processing.	
Secondary storage, similar to main memory, also stores programs and data.	
Section 1.3	
Each computer has its own machine language which is made of streams of 0's and 1's.	
Symbolic languages use mnemonics to represent the various machine language instructions.	
Only the machine and symbolic languages can be understood by computer hardware.	
One benefit to the use of symbolic languages is improved programming efficiency compared to a machine language.	
High-level languages are portable to many different computers.	
The process of converting a high-level language to machine language is known as compilation.	
Section 1.4	
The software used to write programs is known as a text editor.	
A source file contains code written in a programming language that is to be sent to the compiler.	
One purpose of preprocessor commands is to tell the preprocessor to make substitutions in the code.	
The preprocessor prepares the code for translation into machine language.	
Section 1.5	
Careful design of software can result in programs that will be efficient, error-free, and easy to maintain.	
Writing code is the first step in the waterfall model of developing software.	
The waterfall model of developing software places an emphasis on planning and testing.	
Pseudo-code and flowcharts can be used to represent individual algorithms of a program.	
A structure chart can be used to represent the entire program.	
Pseudo-code uses exact programming language syntax to represent a module in the larger program.	
A flowchart uses geometric symbols to represent the logic of a function, or module, in the larger program.	
Except for the most simple program, one set of test data will not completely validate a program.	
Section 1.6	
Software engineering is the use of sound engineering methods and principles to obtain software that is reliable.	
The three constructs of structured programming are: (1) sequence, (2) selection, and (3) looping.	
Section 2.1	
A high-level language allows the programmer to concentrate on the problem at hand and not worry about the specific machine that the program will be using.	
The C programming language is considered a structured programming language.	
The C programming language is not considered a high-level programming language because of its age.	

The following section is from course programming and documentation standards.	
All code found between { and } should be indented two additional spaces.	
Place a single space between all operators and operands.	
Comment all variables to the right of each declaration.	
Declare only one variable per line.	
Select meaningful identifiers (names) for all variables in your program.	
Do not single (or double) space the entire program, use blank lines when appropriate.	
Output of your final `lb01.c` submission must match **EXACTLY** that seen in the example executions.	
There is no need to include example output with your submission.	

The following questions are in regards to Program 2-7 found on pages 71 and 72 of your C programming text:

What is the purpose of the statements on lines 12 through 14?

How are the statements on lines 17 and 18 related in this program?

Can the statements on lines 20 and 21 be moved before those on lines 17 and 18? Why or why not?

Can the statements on lines 20 and 21 be moved after those on lines 23 through 25? Why or why not?

On line 23, what are the `\n` and `%10.2f` **inside** of the quotes accomplishing for the print statement?

On line 25, explain why an additional `\n` should be placed inside of the quotes.

STOP!		**TA Verification:**
Complete Written Problems	Problems from pages 3-4 in this document are complete as the group prepares for programming assignment and quiz at the end of the lab session.	

Lab #1 - Programming Assignment
Due: 30 minutes prior to the start of your next lab meeting.
5 Points Possible

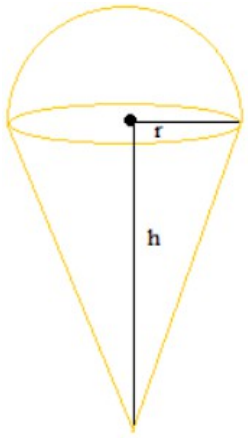
Collaborative Roles for the Lab Session

Collaborative Teaming. For this lab you will be working in your newly assigned teams. If you are unable to complete your assignment during the lab then it is expected that your team meet and collaborate outside of class to finish and submit the problem assigned. **Be sure to exchange contact information during lab today!**

Role:	**Description: Every member will rotate roles at every checkpoint.**
Driver	The driver is in charge of the computer which includes **entering code, saving, testing, and submitting**. This individual should be soliciting the other two members for advice.
Navigator	The role of the navigator is to look over the shoulder of the driver for **syntax errors, logical errors, and concerns related to course standards**. The most common mistakes include failing to pair quotes, misplaced semi-colons, and improper placement of parentheses.
Manager	The manager may not be as close to the driver as the navigator but still plays an important role ensuring that the **algorithm to be implemented is correct** and can be tested using a variety of input to verify correctness.

Problem: Given the height of the right circular cone (h) and radius of the hemisphere (r) in meters, as seen in the image on the right, calculate the volume of the cone, the hemisphere, and the total volume in cubic meters. The final output to generate is the total volume in cubic feet.

- The user will always enter meaningful (positive) measurements. All values should be considered as floating-point data represented by the `float` type.
- Use the symbolic/defined constant for PI as seen in Program 2-7 on pages 71-72 of the C programming text.
- A symbolic/defined constant should also be created to represent the number of cubic feet in a cubic meter (35.315).



Checkpoints During Lab #1:		**TA Verification:**
1 – Getting Started in Vocareum	File `lb01.c` created, assignment header completed, main function inserted, variables declared and commented.	
2 – Input and Calculate	Prompts for user input, accepting user input, and calculation of the volumes and the conversion of the total volume into cubic feet.	
3 – Formatted Output	Display values EXACTLY as seen in the example executions.	
4 – Successful Submission	Save, compile, and test program. Submit and review report related to expected output and course standards.	

Example Execution #1:

```
Enter the height of the cone in meters -> 1
Enter the radius of the hemisphere in meters -> 1

The volume of the hemisphere: 2.09 cubic meters
The volume of the cone: 1.05 cubic meters
The total volume of the shape: 3.14 cubic meters
The total volume of the shape: 110.95 cubic feet
```

Example Execution #2:

```
Enter the height of the cone in meters -> 2.5
Enter the radius of the hemisphere in meters -> 5

The volume of the hemisphere: 261.80 cubic meters
The volume of the cone: 65.45 cubic meters
The total volume of the shape: 327.25 cubic meters
The total volume of the shape: 11556.83 cubic feet
```

Example Execution #3:

```
Enter the height of the cone in meters -> 5
Enter the radius of the hemisphere in meters -> 2.5

The volume of the hemisphere: 32.72 cubic meters
The volume of the cone: 32.72 cubic meters
The total volume of the shape: 65.45 cubic meters
The total volume of the shape: 2311.37 cubic feet
```

Additional Requirements:

1. Add the **lab assignment header** (`vi` shortcut `:hlb` while in command mode) to the top of your program. An appropriate description of your program must be included in the assignment header.
2. **Each of the example executions provided for your reference represents a single execution of the program.** Your program must accept input and produce output **exactly** as demonstrated in the example executions, do not add any "bonus" features not demonstrated in the example executions. Your program will be tested with the data seen in the example executions and an unknown number of additional tests making use of reasonable data.
3. Course standards **prohibit** the use of programming concepts not yet introduced in lecture. For this assignment you can consider all material in the first three chapters of the book, notes, and lectures to be acceptable for use.
4. A program **MUST** compile, be submitted through Vocareum as demonstrated during the lab #0 exercise, and submitted prior to the posted due date to be considered for credit. The C-file you submit must be named exactly: `lb01.c`

Course Programming and Documentation Standards Reminders:

- Indent all code found within the `main` function **exactly** two spaces.
- Place a **single space** between all operators and operands.
- Comment **all** variables to the right of each declaration. Declare only one variable per line.
- Notice that several programs (see program 2-9 on pages 74-75) in the programming text use a single line comment to indicate the start of the local declaration and executable statement sections of the `main` function.
  - At no point during the semester should these two sections ever overlap. You might consider adopting this habit of commenting the start of each section to help you avoid this mistake.
- Select **meaningful identifiers** (names) for all variables in your program.
- Do not single (or double) space the entire program, **use blank lines when appropriate**.