CS 159 – Spring 2022 – Lab #3

Overview of Current Lab:

1. **Important** – submit your attendance at the start of every lab meeting!
2. Please read over the collaborative teaming expectations.
3. **Work with your team on the written problems found in this document.** These problems are the basis for the lab quiz and may include content not introduced in lecture.
4. Begin your work on the programming assignment.
5. Complete the Collaborative Group Communication form and submit to your lab instructor for review.
6. During the final 10 minutes of your lab today you will complete the second 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 #3 – 5 points possible: The lab quiz will be made available on Brightspace (Week #4 module) during the final 10 minutes of your lab today. The quiz will emphasize material from chapter 3 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 #3 – 5 points possible: Every lab programming assignment is due 30 minutes prior to the start of your next lab meeting. Pages 93-94 of the course notes packet provide examples of selection by calculation problems.

- **How might collaboration be useful on this particular programming assignment?** Working out the math and in turn the logic necessary to solve this problem is going to be a paper and pencil exercise that is likely to consume more time than the act of coding the solution.

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 3 and the course standards.

Statement	True / False
The coding assignment for lab #3 forbids the use of selection constructs (if, else if, else, switch), logical operators (AND, OR, NOT), and relational operators (<, <=, >, >=, ==, !=).	**TRUE**
All of the compound assignment operators share the same level of operator precedence.	
Operator precedence can be used to bind operators to operands and from that will determine the order of operations in an expression.	
A single-type operation will generate a result of that same type.	
The use of a precision modifier when displaying a floating-point value will result in truncating all digits beyond the specified precision value.	
Converting a higher ranked data type to that of a lower ranked data type may result in the loss of data.	
Section 3.1	
When two operators with the same precedence occur in an expression and their associativity is left-to-right, the left operator is evaluated first.	
An expression always reduces to a single value.	
The value of the postfix increment expression is determined before the variable is increased.	
The operand in a postfix or prefix expression must be a variable.	
The assignment expression evaluates the operand on the right side of the operator and places its value in the variable on the left.	
When a compound assignment is used with an expression, the expression is only evaluated first when parentheses are used to raise its level of precedence. [Example: `x *= (a + b)`]	
The compound assignment operator (/=) has a higher level of precedence then the addition operator (+).	
Section 3.4	
The result of an expression is undefined when it attempts to modify a single variable more than once.	
Section 3.5	
When the types of two operands in a binary expression are different, C automatically converts one type to the other.	
When the types of two operands in a binary expression (excluding assignment operators) are different, the lower-ranked type is promoted to the rank of the higher type.	
In an assignment statement, promotion occurs if the right expression has a lower rank than the variable on the left and demotion occurs if the right expression has a higher rank.	
An explicit type conversion is the programmer taking control and determining the data type of an operand in an expression.	
To cast data from one type to another, we specify the new type in parentheses before the value we want converted.	
Section 3.6	
The defined constant results in an automatic substitution of the value that follows the symbol where it is found in the program. One exception, no such substitution will take place inside of quotes.	
Problems associated with defined constants are difficult to resolve as the programmer views the original statement and not the statement with the error after the substitution takes place.	
Section 3.7	
One programming technique to simplify code is to use parentheses, even when unnecessary.	

Solve the following problems related to standard library constants and functions:

In which library will you find the `abs` function? How is this function different from the `fabs` function found in `math.h`?

What is `INT_MAX`? Where is it defined? What are `FLT_MAX` and `FLT_DIG`? Where are these defined?

Identify each of the constants found `math.h` below:

`M_E`

`M_LOG10E`

`M_PI`

`M_LN2`

`M_LOG2E`

`M_LN10`

Solve the following problems related to material found in Chapter 3.

- 25 % 11
- 132 % 2
- 134 % 2
- 16 % 17
- 49 % 5
- 133 % 2
- 135 % 2
- 2 % 1

Given A and B are integer variables, both are greater than zero, and A is greater than B. What is the result of the expression below?

- B % A

Given A and B are integer variables, both are greater than zero, and A is less than B. What is the range of values possible as a result of the expression below?

- B % A

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 #3 - 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 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,** can be tested using a variety of input to verify correctness, and **complies with the additional requirements of the assignment.**

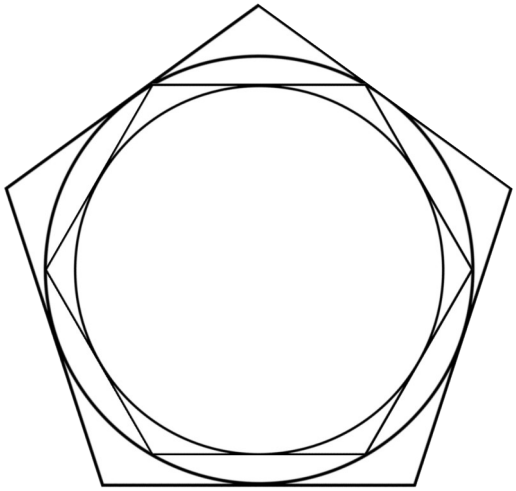
Problem: In the image given there is a smaller circle inscribed inside of a regular HEXAGON. The hexagon is inscribed inside of a larger circle. And finally the larger circle is inscribed inside of a regular PENTAGON.

Given either the inradius or circumradius of the pentagon calculate and display the following:

- Apothem, base, and area of the pentagon.
- Area of the large, hexagon, and small circle.

Requirements of your solution:

- All code will continue to be placed within the `main` function.
- Any use of logical operators, relational operators, `bool` variables, or selection constructs is **prohibited**.
- A violation of either of the above will result in your assignment being awarded no points.



Checkpoints During Lab #3:		**TA Verification:**
1 – Plan the Mathematics	Revisit lab #2 to identify those expressions that need to be redesigned and those that only require to be rearranged in the logic. Determine how selection by calculation will be utilized.	
2 – Getting Started in Vocareum	File `lb03.c` created, assignment header completed, `main` function inserted, variables declared and commented.	
3 – Input and Calculate	Prompts for user input, accepting user input, and necessary calculations.	
4 – Formatted Output	Display all output **EXACTLY** as seen in the example executions.	
5 – Successful Submission	Save, compile, and test program. Submit and review report related to expected output and course standards.	

Example Execution #1:

```
Select the (I)ncircle or (C)ircumcircle radius -> C
Enter the length -> 69

-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
Apothem of pentagon:        55.82
Base of pentagon:           81.11
Area of pentagon:        11319.95
Area of large circle:     9789.56
Area of hexagon:          8095.90
Area of small circle:     7342.17
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
```

Example Execution #2:

```
Select the (I)ncircle or (C)ircumcircle radius -> I
Enter the length -> 21.5

-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
Apothem of pentagon:        21.50
Base of pentagon:           31.24
Area of pentagon:         1679.22
Area of large circle:     1452.20
Area of hexagon:          1200.96
Area of small circle:     1089.15
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
```

Example Execution #3:

```
Select the (I)ncircle or (C)ircumcircle radius -> C
Enter the length -> 16.7

-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
Apothem of pentagon:        13.51
Base of pentagon:           19.63
Area of pentagon:          663.10
Area of large circle:      573.45
Area of hexagon:           474.24
Area of small circle:      430.09
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
```

Example Execution #4:

```
Select the (I)ncircle or (C)ircumcircle radius -> I
Enter the length -> 14

-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
Apothem of pentagon:        14.00
Base of pentagon:           20.34
Area of pentagon:          712.01
Area of large circle:      615.75
Area of hexagon:           509.22
Area of small circle:      461.81
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
```

Reminder from the problem description:

- All code will continue to be placed within the `main` function.
- Any use of logical operators, relational operators, `bool` variables, or selection constructs is **prohibited**.
- A violation of either of the above will result in your assignment being awarded no points.

Additional Requirements:

1. Add the **lab assignment header** (`vi` shortcut `:hlb` while in command mode) to the top of your program. An appropriate description the logic 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.
    - The user of the program will always enter `I` or `C` as the character input.
    - All floating-point variables must be of the `double` type.
    - Use the constant value `M_PI` where needed for the constant pi.
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.
    - Any use of logical operators, relational operators, `bool` variables, or selection constructs is **prohibited** and would violate requirements of this assignment **resulting in no credit being awarded for your effort.**
    - **All code for this assignment will be placed inside of the `main` function.** User-defined functions will be a requirement beginning with lab #4.
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: `lb03.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**.
- Maximize your use of symbolic/defined constants and minimize your use of literal constants.