

## CS 159 – Spring 2021 – Lab #6

**What will you submit?** A single C-file will be submitted electronically via the guru server. An example submission was conducted as part of the Account Configuration Activity. If you have a concern regarding how to submit work, please **contact** course staff **prior** to the deadline for this, and all, assignments. The programming assignment is due on Friday March 12, 2021 at 11:00pm (LOCAL WEST LAFAYETTE, IN TIME). **No late work will be accepted.**

---

### Weekly Quiz #6:

The weekly quiz will be available (Week 8 module on Brightspace) until the same date and time that the programming assignment is due. It is strongly recommended that you complete the attached problems, the programming assignment, and watch all relevant lectures before attempting the quiz.

The quiz will emphasize chapter 5 material, the written problems in this document, the lab programming assignment, and the course programming and documentation standards as used in this lab. 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. Each quiz has a 15-minute time limit.

---

### Collaborative Teaming:

- **How do I know who is on my lab team?**
  - **On-campus students:** TAs have contacted their students via e-mail.
  - **On-line students:** Visit the Start Here module on Brightspace and locate the Distance Learning Team Assignment spreadsheet.
- **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.
- **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. Other on-line tools may come in handy when trying to collaborate on specific segments of code, just make sure they protect your code from being posted publicly! One popular service from previous semesters is [codeshare.io](https://codeshare.io).
- **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. **Only one person per group will make submissions for the entire group.** The grader for your section cannot be expected to grade submissions from multiple members of the same group to determine which submission you actually want graded.
  - In order for each member of the group to get credit for the programming problem associated with this lab, their **Purdue University e-mail address must appear in the assignment header.**
- **How might collaboration be useful on this particular programming assignment?**
  - This is the last lab before lab exam #2. Based on your experience with the first lab exam you may safely assume that the second lab exam will have a problem related to labs #4, #5, and this lab #6. Share in your early communication the importance of a well designed solution in order to make modifications for the second lab exam easier.
  - Lab #6 represents the first time this semester that the use of chapter 5 (selection) is permitted. The problem of selection is not nearly what it is on the fourth homework assignment but the comparison of two floating-point values for equality requires more than, and will likely not use, the == operator.
  - The March 9<sup>th</sup> midterm will not cover selection but will cover user-defined functions, this lab will have many.

**(Task #1) - Solve the following problems related to material found in Chapter 5 and the course standards.**

Statement	True or False
The use of arrays, including character pointers to represent string data, or any technique of repetition would violate requirements of this assignment and result in no credit being awarded for your effort.	<b>TRUE</b>
<b>Section 5.1</b>	
A piece of data is called logical if it conveys the idea of true or false.	
C programmers use other types, such as integers, to represent logical data.	
If a data value is zero it is considered false, but any non-zero value is considered true.	
The logical AND (&&) operator is true only when both operands are true.	
The logical OR (  ) operator is true only when exactly one of its operands is true.	
The AND and OR operator share the same level of operator precedence.	
The short-circuit method of evaluating logical expressions will stop evaluating the current expression as soon as the result can be determined.	
The complement of the equal operator is the not equal operator.	
The complement of the greater than operator is the less than operator.	
When attempting to print the result of a logical expression that is true as an integer the result will always be 1.	
<b>Section 5.2</b>	
The logical expression of an <code>if...else</code> construct must be enclosed in parentheses.	
There is no semi-colon that follows the logical expression of an <code>if...else</code> construct.	
The statements found inside of an <code>if...else</code> may be any statement, including another <code>if...else</code> construct.	
The expressions <code>if(a != 0)</code> and <code>if(!a)</code> are complements.	
The expressions <code>if(a == 0)</code> and <code>if(a)</code> are complements.	
The dangling <code>else</code> logical error can be corrected by indenting the <code>if</code> and <code>else</code> the same number of spaces.	
The conditional expression has three operands and a two-token operator.	
Conditional expressions cannot be nested as the resulting code becomes too complex.	
<code>3 &amp;&amp; -3 &amp;&amp; 10 &amp;&amp; -10</code>	
<code>3    -3    10    -10</code>	
<code>3    6 &amp;&amp; 0</code>	
<code>3 == 4    6    8</code>	
<code>3 == 3 &amp;&amp; -1 &amp;&amp; 1</code>	
<code>6 % 2    7 % 2</code>	

**Task #1 continued - Solve the following problems related to material found in Chapter 5 and the course standards.**

Statement	True / False
The following two logical expressions are equivalent for all integer ( <code>int</code> ) values of <code>x</code> : <code>!(x &lt; 10) and x &gt;= 10</code>	
The following two logical expressions are equivalent for all non-negative integer ( <code>int</code> ) values of <code>x</code> : <code>x % 2 and x % 2 != 0</code>	
The complement of <code>x % 3 == 0    x % 3 == 2</code> is <code>x % 3 != 0 &amp;&amp; x % 3 != 2</code> for all non-negative integer ( <code>int</code> ) values of <code>x</code> .	
The complement of <code>x &gt; 0 &amp;&amp; x &lt; 10    y + 2 == 0</code> is <code>x &lt;= 0    x &gt;= 10 &amp;&amp; y + 2 != 0</code>	
The compiler will issue a warning when an assignment operator rather than the equality operator is used as the logical expression of an <code>if</code> condition.	
It is a course standard to make use of <code>{</code> and <code>}</code> with all <code>if-else</code> constructs.	
It is a course standard to indent all code within the body of a selection construct two additional spaces.	

**Identify a value of integer (`int`) variables `x` and `y` that demonstrate the following expressions are NOT complements:**

`x > 0 && x < 10 || y != 0`

`x <= 0 || x >= 10 && y == 0`

## Lab #6 – Programming Assignment

**Due:** Friday March 12, 2021 at 11:00pm (time local to West Lafayette, IN)

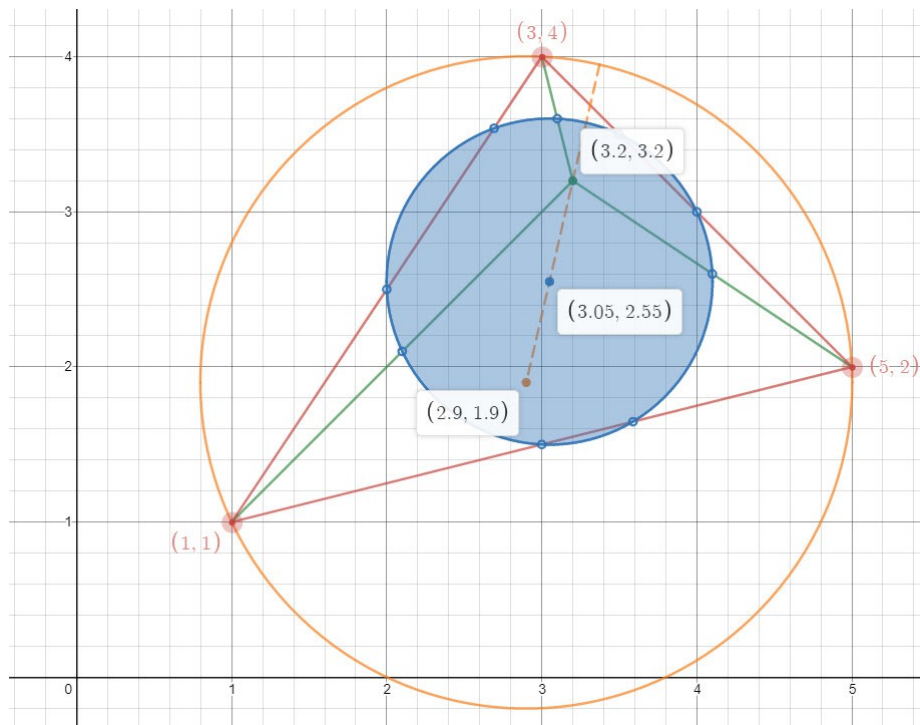
### 10 Points Possible

**Problem:** Given three points (A, B, C) that will always create a triangle with a positive area, calculate the location of the orthocenter, circumcenter, and the location of the center of the nine point circle and whether that point is inside or outside of the triangle.

How to determine whether the nine point circle center is inside or outside of the triangle?

1. Calculate the area of the three triangles created by (1) points A, B, and the nine point circle center, (2) points B, C, and the nine point circle center, and (3) points C, A, and the nine point circle center.
2. If the total of the three areas is equal to that of the triangle created by points A, B, and C, then the nine point circle center lies inside of the triangle, **otherwise it lies outside of the triangle**. Comparing two floating-point values for an EXACT match is prone error, for this assignment if the absolute value of the difference between the area of the larger triangle and the total area of the three smaller triangles is less than 0.001 then the values will be considered equivalent.

For this assignment you will be **required** to implement the user-defined functions (from chapter 4).



### Example Execution #1:

```
Enter X coordinate #1 -> 1
Enter Y coordinate #1 -> 1
Enter X coordinate #2 -> 5
Enter Y coordinate #2 -> 2
Enter X coordinate #3 -> 3
Enter Y coordinate #3 -> 4
```

Coordinates of orthocenter: 3.20, 3.20

Coordinates of circumcenter: 2.90, 1.90

The nine point circle center: 3.05, 2.55 is inside of the triangle.

### Example Execution #2:

```
Enter X coordinate #1 -> 3
Enter Y coordinate #1 -> 3
Enter X coordinate #2 -> -4
Enter Y coordinate #2 -> -1
Enter X coordinate #3 -> 3
Enter Y coordinate #3 -> 5
```

Coordinates of orthocenter: 6.43, -1.00  
Coordinates of circumcenter: -2.21, 4.00  
The nine point circle center: 2.11, 1.50 is outside of the triangle.

### Example Execution #3:

```
Enter X coordinate #1 -> 4.5
Enter Y coordinate #1 -> -3.5
Enter X coordinate #2 -> -6.1
Enter Y coordinate #2 -> 6.5
Enter X coordinate #3 -> -5.5
Enter Y coordinate #3 -> -0.5
```

Coordinates of orthocenter: -9.46, -4.70  
Coordinates of circumcenter: 1.18, 3.60  
The nine point circle center: -4.14, -0.55 is inside of the triangle.

### Example Execution #4:

```
Enter X coordinate #1 -> 10.3
Enter Y coordinate #1 -> -7
Enter X coordinate #2 -> 1.6
Enter Y coordinate #2 -> -2.4
Enter X coordinate #3 -> 8.4
Enter Y coordinate #3 -> -3.2
```

Coordinates of orthocenter: 11.42, 2.51  
Coordinates of circumcenter: 4.44, -7.55  
The nine point circle center: 7.93, -2.52 is outside of the triangle.

---

**All course programming and documentation standards are in effect for this and each assignment this semester. Please review this document!**

---

### Additional Requirements:

1. Add the **lab assignment header** (vi shortcut h1b while in command mode) to the top of your program. An appropriate description of your program must be included in the assignment header. Include the Purdue University e-mail addresses of **each contributing group member** 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. 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**. All floating-point variables must be of the `double` type.
3. For this assignment you will be **required** to implement the user-defined functions (from chapter 4). Failing to follow course standards as they relate to good user-defined function use will result in a **zero for this assignment**.

### Additional Requirements (continued):

4. Revisit **course standards as it relates what makes for good use of user-defined functions, what is acceptable to retain in the main function, and when passing parameters by address is appropriate.**
  - In many cases user-defined function use should result in a `main` function that only declares variables and makes calls functions.
5. Course standards **prohibit** the use of programming concepts not yet introduced in lecture. For this assignment you can consider all material in the first FIVE chapters of the book, notes, and lectures to be acceptable for use.
  - Each of the available selection constructs, logical operators, and relational operators are available for use in this assignment. **Do not** make use of the `bool` data type due to the extra include required.
  - The use of arrays, including character pointers to represent string data, or any technique of repetition would violate requirements of this assignment and **result in no credit being awarded for your effort.**
6. A program **MUST** compile, be submitted through the `guru` server prior to the posted due date to be considered for partial credit. The C-file you submit must be named exactly: `lab06.c`

### Course Programming and Documentation Standards Reminders:

- Code found inside the body of a selection construct must be indented two additional spaces.
- Make use of `{` and `}` with all relevant selection constructs.
- See page 258 of your C programming text regarding the proper indentation for a `switch` construct.
- Use the course function header (`head_fx` vi shortcut `hfx` while in command mode) for every user-defined function in your program.
  - List and comment **all parameters** to a function, one per line, in the course function header.
  - **All function declarations** will appear in the global declaration section of your program.
  - **The user-defined function definitions will appear in your program after the main function.**
- Maximize your use of symbolic/defined constants and minimize your use of literal constants.
- Indent all code found within the `main` and all user-defined functions **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.**
- There is no need to include example output with your submission.

### Auto-Grade Tool

- We have implemented what is being referred to as the auto-grade tool. At the time of a successful assignment submission you may receive some feedback on your program in regards to course programming and documentation standards. This feedback may include a potential deduction that you will receive once your assignment is reviewed by your grader.
- It is expected that graders verify those notes identified by this tool to ensure that they are indeed applicable and reasonable to the submission. Graders may make additional deductions for those standards not identified by the new tool.
- We hope that this feedback helps with the enforcement of course standards, consistency in grading across sections, and to encourage students to revise their work when problems are identified before the assignment deadline passes. It is possible to resubmit an assignment for grading up to the advertised deadline. Only the final successful submission is retained and evaluated.

**Helpful External Resources:**

- <https://byjus.com/circumcenter-calculator/>
- <https://byjus.com/orthocenter-calculator/>
- [https://en.wikipedia.org/wiki/Nine-point\\_circle](https://en.wikipedia.org/wiki/Nine-point_circle)
- <https://www.desmos.com/calculator/tzihmpwt3v>