

LAB 3 DEBUGGING DOCUMENT

YOUR NAME: *Zavier Fowler and Malachi Dorman*

PART 1: DEBUG A GIVEN PROGRAM

This guide will walk you through the steps of iterative debugging. Use this document to track errors, solutions, and screenshots of your work. Please follow the instructions in this document to track your errors, document your solutions, answer questions, and paste screenshots in this document as instructed. **You will be uploading this completed document with your submission for this lab assignment.**

STEP 1: WHAT IS THE PROGRAM SUPPOSED TO DO?

Write down the overall goal of the given **lab3a.cpp** program and the steps it should take to achieve that goal.

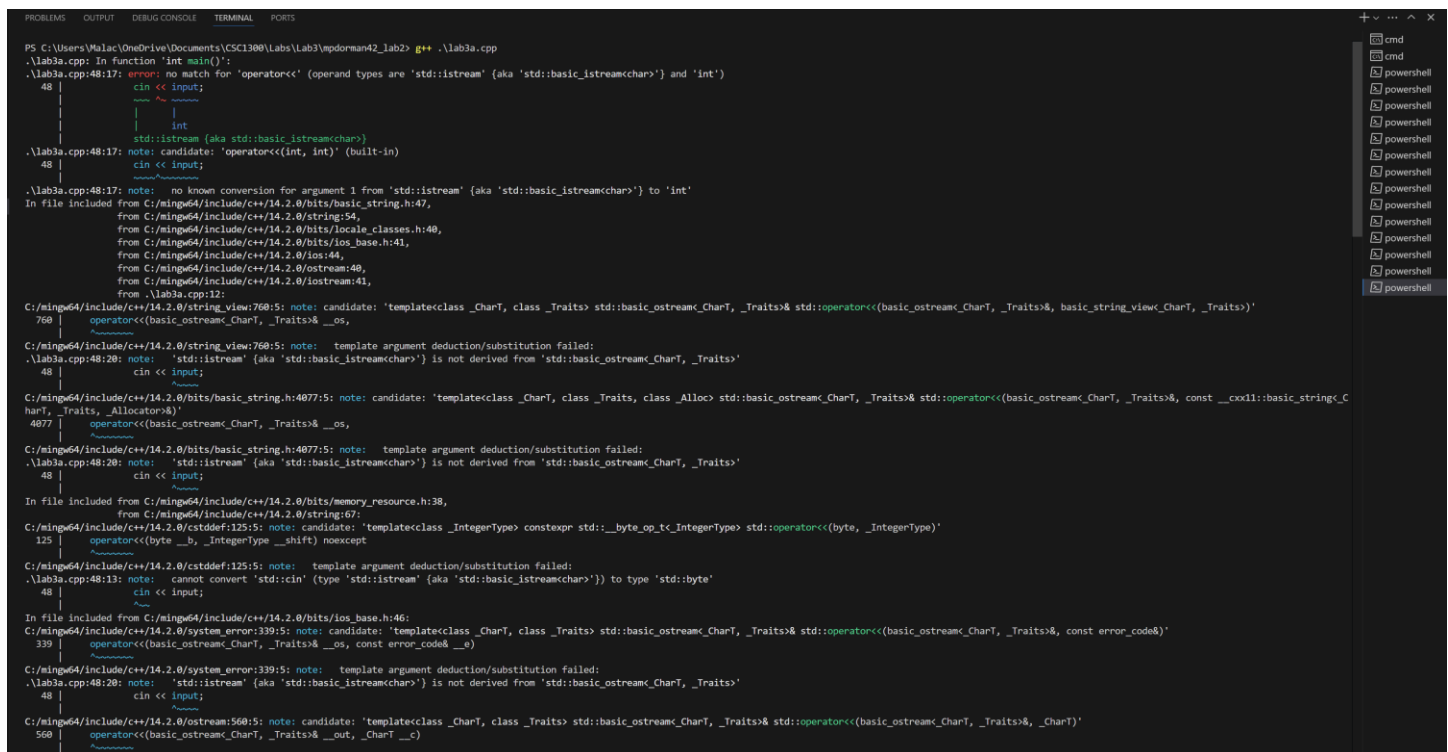
Note: The **sample output** of the corrected version of lab2a.cpp was provided in **Part 1** of the **Spring 2025 Lab 2 Assignment** document.

The program should ...

STEP 2: COMPILE AND RUN THE PROGRAM

Record the observed behavior and output. Include a screenshot of the output here:

Screenshot 1: *(Paste screenshot)*



```
PS C:\Users\MalachOneDrive\Documents\CSC1300\Labs\Lab3\mpdorman42_lab2> g++ .\lab3a.cpp
.\lab3a.cpp: In function 'int main()':
.\lab3a.cpp:48:17: error: no match for 'operator<<' (operand types are 'std::istream' (aka 'std::basic_istream<char>') and 'int')
   48 |         cin << input;
      |         ~~~~^~~~~~
      |         |
      |         int
      |         |
      |         std::istream (aka std::basic_istream<char>)
.\lab3a.cpp:48:17: note: candidate: 'operator<<(int, int)' (built-in)
   48 |         cin << input;
      |         ~~~~^~~~~~
.\lab3a.cpp:48:17: note: no known conversion for argument 1 from 'std::istream' (aka 'std::basic_istream<char>') to 'int'
In file included from C:/mingw64/include/c++/14.2.0/bits/basic_string.h:47,
                 from C:/mingw64/include/c++/14.2.0/string:54,
                 from C:/mingw64/include/c++/14.2.0/locale_classes.h:40,
                 from C:/mingw64/include/c++/14.2.0/bits/ios_base.h:41,
                 from C:/mingw64/include/c++/14.2.0/ios:44,
                 from C:/mingw64/include/c++/14.2.0/ostream:40,
                 from C:/mingw64/include/c++/14.2.0/iostream:41,
                 from .\lab3a.cpp:12:
C:/mingw64/include/c++/14.2.0/string_view:760:5: note: candidate: 'template<class CharT, class Traits> std::basic_ostream<CharT, Traits>& std::operator<<(basic_ostream<CharT, Traits>&, basic_string_view<CharT, Traits>)'
   760 |     operator<<(basic_ostream<CharT, Traits>& __os,
      |     ~~~~~^~~~~
C:/mingw64/include/c++/14.2.0/string_view:760:5: note: template argument deduction/substitution failed:
.\lab3a.cpp:48:20: note:   'std::istream' (aka 'std::basic_istream<char>') is not derived from 'std::basic_ostream<CharT, Traits>'
   48 |         cin << input;
      |         ~~~~^~~~~~
C:/mingw64/include/c++/14.2.0/bits/basic_string.h:4077:5: note: candidate: 'template<class CharT, class Traits, class Alloc> std::basic_ostream<CharT, Traits>& std::operator<<(basic_ostream<CharT, Traits>&, const __cxx11::basic_string<C
harT, Traits, Allocator>&)'
  4077 |     operator<<(basic_ostream<CharT, Traits>& __os,
      |     ~~~~~^~~~~
C:/mingw64/include/c++/14.2.0/bits/basic_string.h:4077:5: note: template argument deduction/substitution failed:
.\lab3a.cpp:48:20: note:   'std::istream' (aka 'std::basic_istream<char>') is not derived from 'std::basic_ostream<CharT, Traits>'
   48 |         cin << input;
      |         ~~~~^~~~~~
In file included from C:/mingw64/include/c++/14.2.0/bits/memory_resource.h:38,
                 from C:/mingw64/include/c++/14.2.0/string:67:
C:/mingw64/include/c++/14.2.0/cstdint:125:5: note: candidate: 'template<class IntegerType> constexpr std::byte op_<<(IntegerType) std::operator<<(byte, IntegerType)'
   125 |     operator<<(byte __b, IntegerType __shift) noexcept
      |     ~~~~~^~~~~
C:/mingw64/include/c++/14.2.0/cstdint:125:5: note: template argument deduction/substitution failed:
.\lab3a.cpp:48:13: note:   cannot convert 'std::cin' (type 'std::istream' (aka 'std::basic_istream<char>')) to type 'std::byte'
   48 |         cin << input;
      |         ~~~~^~~~~~
In file included from C:/mingw64/include/c++/14.2.0/bits/ios_base.h:46:
C:/mingw64/include/c++/14.2.0/system_error:339:5: note: candidate: 'template<class CharT, class Traits> std::basic_ostream<CharT, Traits>& std::operator<<(basic_ostream<CharT, Traits>&, const error_code&)'
   339 |     operator<<(basic_ostream<CharT, Traits>& __os, const error_code& __e)
      |     ~~~~~^~~~~
C:/mingw64/include/c++/14.2.0/system_error:339:5: note: template argument deduction/substitution failed:
.\lab3a.cpp:48:20: note:   'std::istream' (aka 'std::basic_istream<char>') is not derived from 'std::basic_ostream<CharT, Traits>'
   48 |         cin << input;
      |         ~~~~^~~~~~
C:/mingw64/include/c++/14.2.0/ostream:569:5: note: candidate: 'template<class CharT, class Traits> std::basic_ostream<CharT, Traits>& std::operator<<(basic_ostream<CharT, Traits>&, const CharT&)'
   569 |     operator<<(basic_ostream<CharT, Traits>& __out, CharT __c)
      |     ~~~~~^~~~~~
```

STEP 3: DOCUMENT & FIX SYNTAX ERRORS

Note: Debugging is an iterative process; it's normal to encounter new issues as you fix others. Don't hesitate to ask peers or instructors for help if you're stuck.

For each syntax error:

1. Locate the error:

- Document any compilation errors.
- Read the error message and identify the line number.
- Look for red squiggles or warnings in your code editor.

2. Understand the error:

- Hover over the error message or search online to understand what it means.
- Document the meaning of the error here:

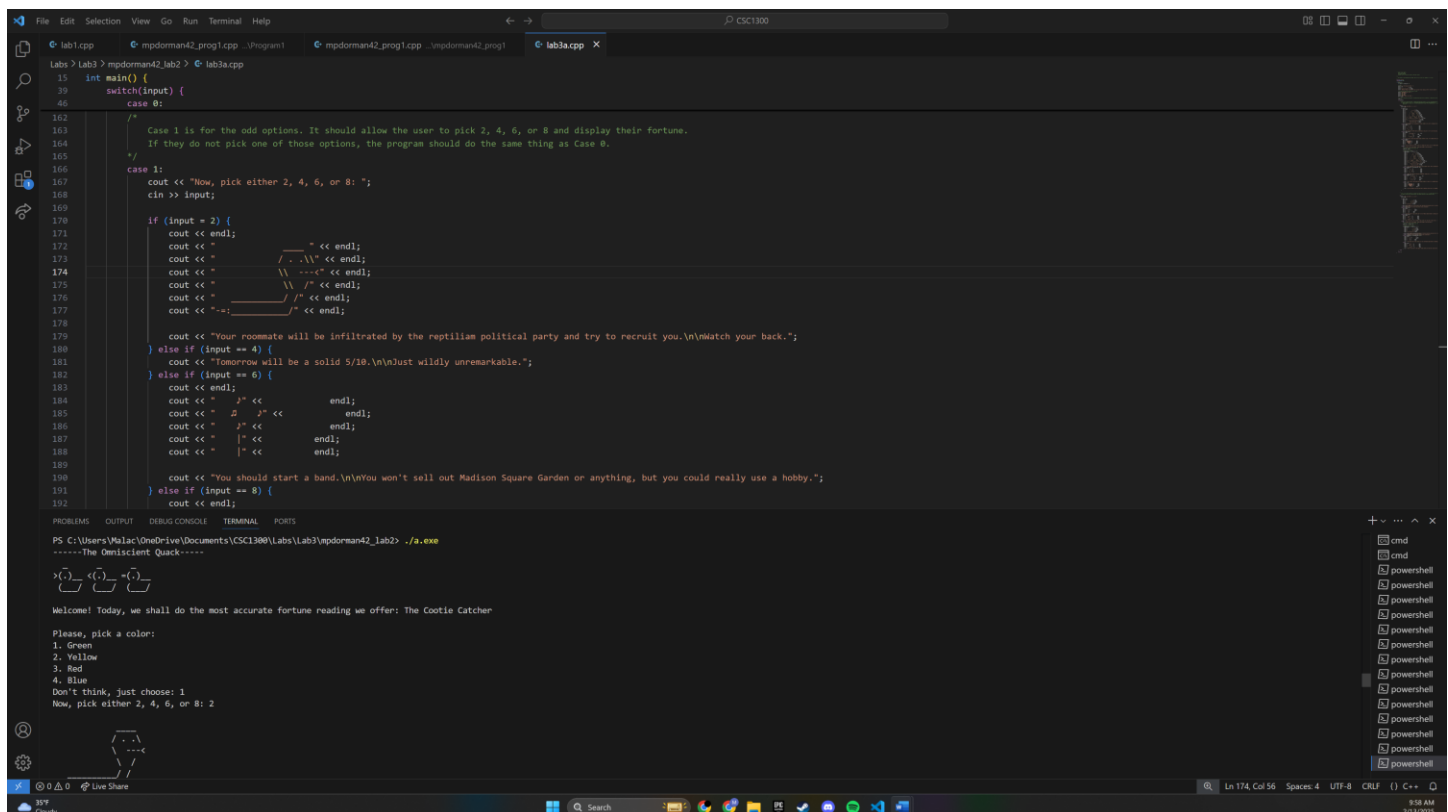
Error Explanation: *(Write explanation)*

3. Fix the error:

- Implement a solution and recompile the program.
- Record whether the error went away.

4. Repeat until all syntax errors are resolved, adding a new Error Explanation and screenshot for each error you fix.

Screenshot 2: *(Paste updated screenshot)*



```
15 int main() {
16     switch(input) {
17     case 0:
18         // Case 1 is for the odd options. It should allow the user to pick 2, 4, 6, or 8 and display their fortune.
19         // If they do not pick one of those options, the program should do the same thing as Case 0.
20         // Case 1:
21         cout << "Now, pick either 2, 4, 6, or 8: ";
22         cin >> input;
23
24         if (input == 2) {
25             cout << endl;
26             cout << "      " << endl;
27             cout << "      / . .\\ " << endl;
28             cout << "      \\ ----> " << endl;
29             cout << "      // " << endl;
30             cout << "      / " << endl;
31             cout << "      " << endl;
32
33             cout << "Your roommate will be infiltrated by the reptilian political party and try to recruit you.\n\nMatch your back.";
34         } else if (input == 4) {
35             cout << "Tomorrow will be a solid 5/10.\n\nJust wildly unremarkable.";
36         } else if (input == 6) {
37             cout << endl;
38             cout << "  2" << endl;
39             cout << "  2  2" << endl;
40             cout << "  2" << endl;
41             cout << "  " << endl;
42
43             cout << "You should start a band.\n\nYou won't sell out Madison Square Garden or anything, but you could really use a hobby.";
44         } else if (input == 8) {
45             cout << endl;
46         }
47     }
48 }
```

PS C:\Users\Malac\OneDrive\Documents\CS1300\Labs\Lab3\mpdorman42_lab2> ./a.exe

-----The Omiscient Quack-----

Welcome! Today, we shall do the most accurate fortune reading we offer: The Cootie Catcher

Please, pick a color:

1. Green

2. Yellow

3. Red

4. Blue

Don't think, just choose: 1

Now, pick either 2, 4, 6, or 8: 2

Fixed incorrect ">>" for cin and added a missing semicolon.

STEP 4: ADDRESS LOGIC ERRORS

If the program runs but the output is incorrect:

1. Identify the problem:

- Compare the program's actual output to the expected output.
- Describe what the program is doing versus what it should be doing.

2. **Observed Behavior:** *(Write here) Only option 1 works in the first input. Then no matter what it will display input 2 and print it twice.*

3. **Expected Behavior:** *(Write here)*

4. Test individual sections:

- Use print/cout statements or a debugger to isolate which part of the code is not working as intended.

5. Locate and fix the error:

- Analyze the problematic section.
- Document the issue and your solution:

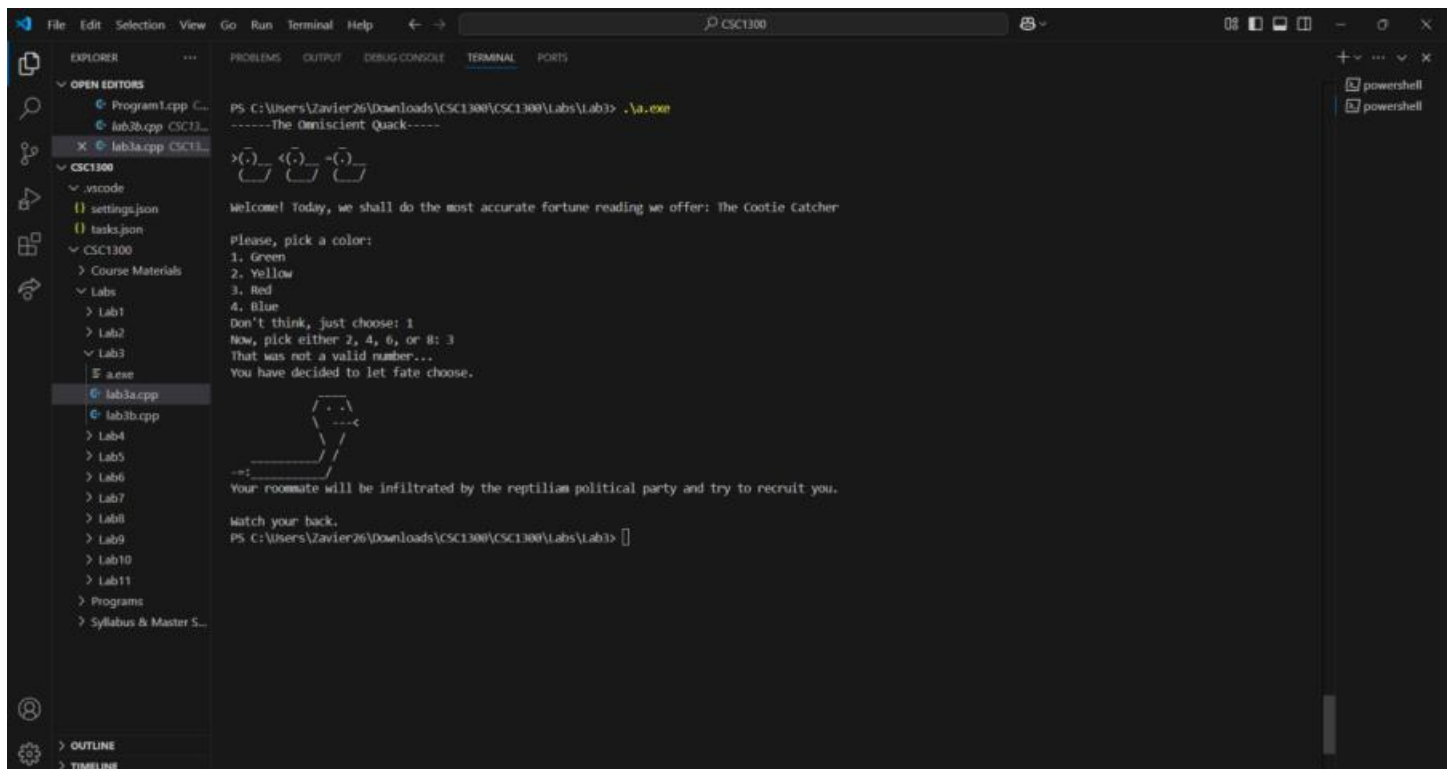
6. **Logic Error:** It would put out the same out every single time

7. **Solution:** Add a srand(time(0)) to create an actual random number generator

8. Re-run the program:

- Test whether your fix resolved the issue.
- If not, repeat this step. **Do not** delete your previous responses or screenshot, just add to them with the new bug and new solution.

Screenshot 3:



The screenshot shows a Visual Studio Code editor with a C++ project named 'CSC1300'. The Explorer pane on the left shows the file structure, including 'lab3a.cpp' and 'lab3b.cpp'. The main editor area displays the code for 'lab3a.cpp'. The Output pane on the right shows the program's execution output. The output includes a welcome message, a list of color options (Green, Yellow, Red, Blue), a prompt to pick a color, and a message about a reptilian political party. The program appears to be running in a terminal window.

STEP 5: ITERATE UNTIL CORRECT

1. Continue debugging:

a. Repeat each debugging process until the program runs correctly and produces the desired output.

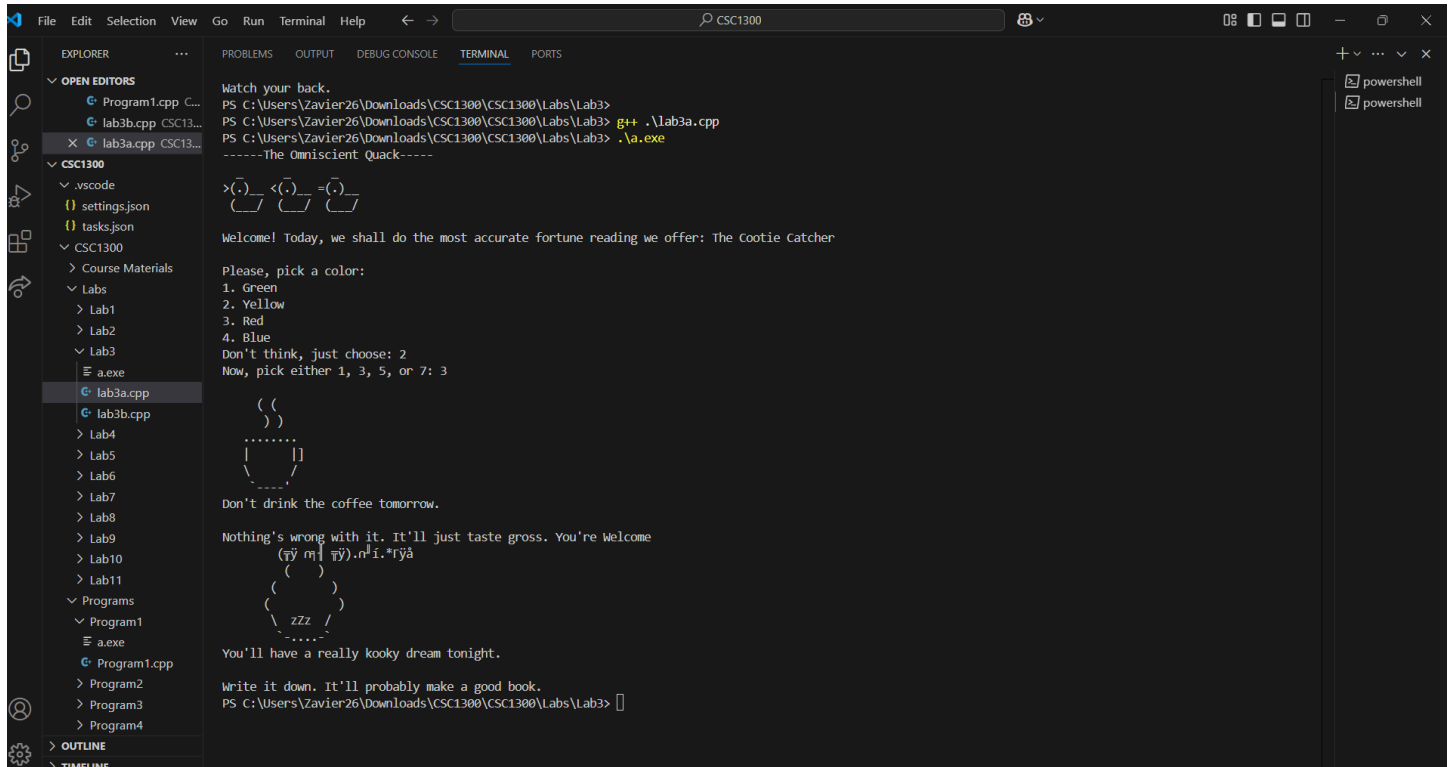
2. Document additional issues and fixes:

a. Use the template from Steps 3 and 4 to track any new errors introduced during debugging.

3. Final Output:

a. Paste a screenshot of the final, correct output below:

Screenshot 4:



PART 2: PRACTICE ON YOUR OWN

First, go back to the **Spring 2025 Lab 3 Assignment** document and read through **Part 2** assignment direction that guides you on creating **lab3b.cpp**. Then, come back to this document to answer the questions and provide screenshots while following the directions.

STEP 1: DEFINE THE PROGRAM'S GOAL

What is the program supposed to do?

- The program should ... Generate a random uppercase character
- Take in the user input and make sure that it is uppercase (hint: look at the character conversion functions)
- See if the user guessed the character correctly. If not, it should output if the user was within 5 characters of the generated character.

If you are doing Option 2, this is where you will include your planned test cases for your program. Think about what inputs you will need and how the output should change depending on what the user enters for those inputs.

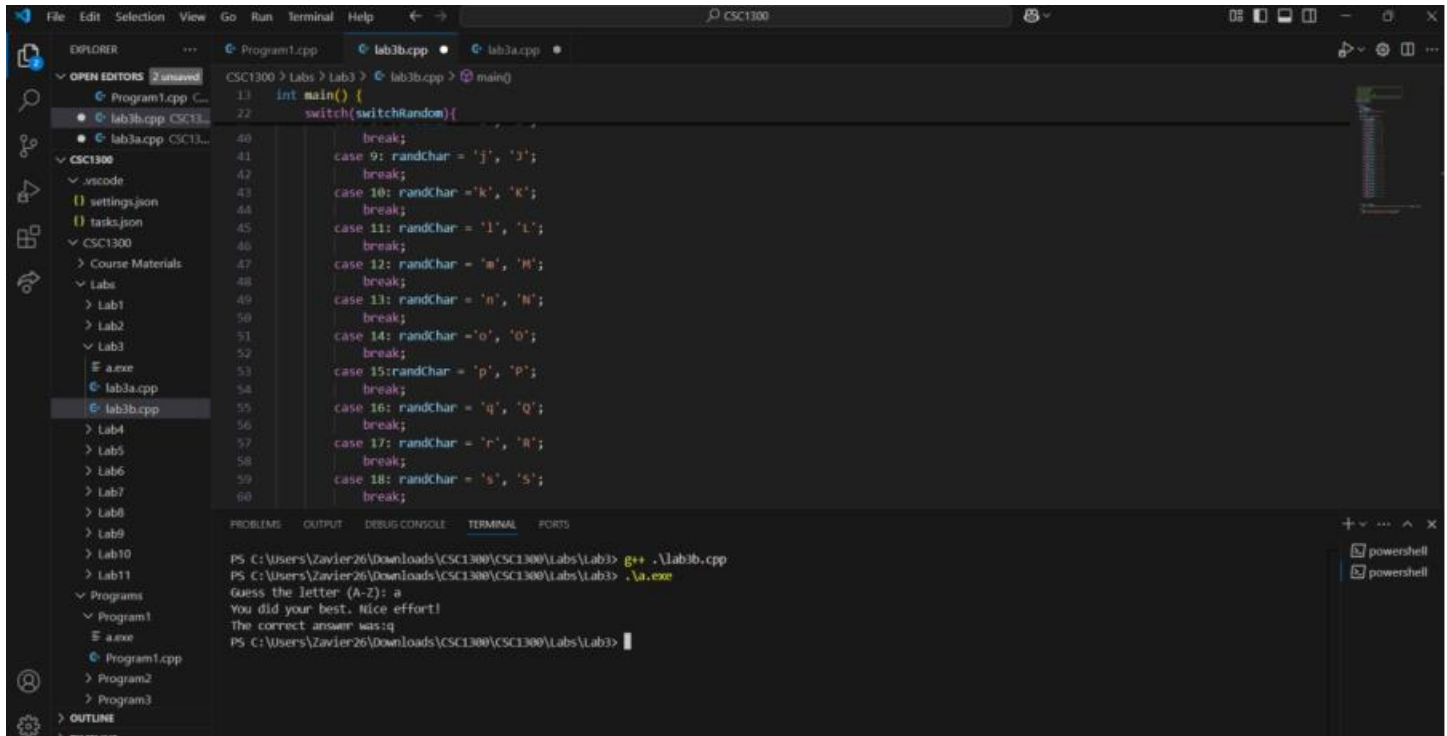
STEP 2: RUN YOUR PROGRAM

Compile and run the program.

You may do this step multiple times before the full program is finished, in fact, this is encouraged! This can help you catch bugs earlier on and helps you know which parts of your program are already working as intended.

Include a screenshot of the output here:

Screenshot 5:

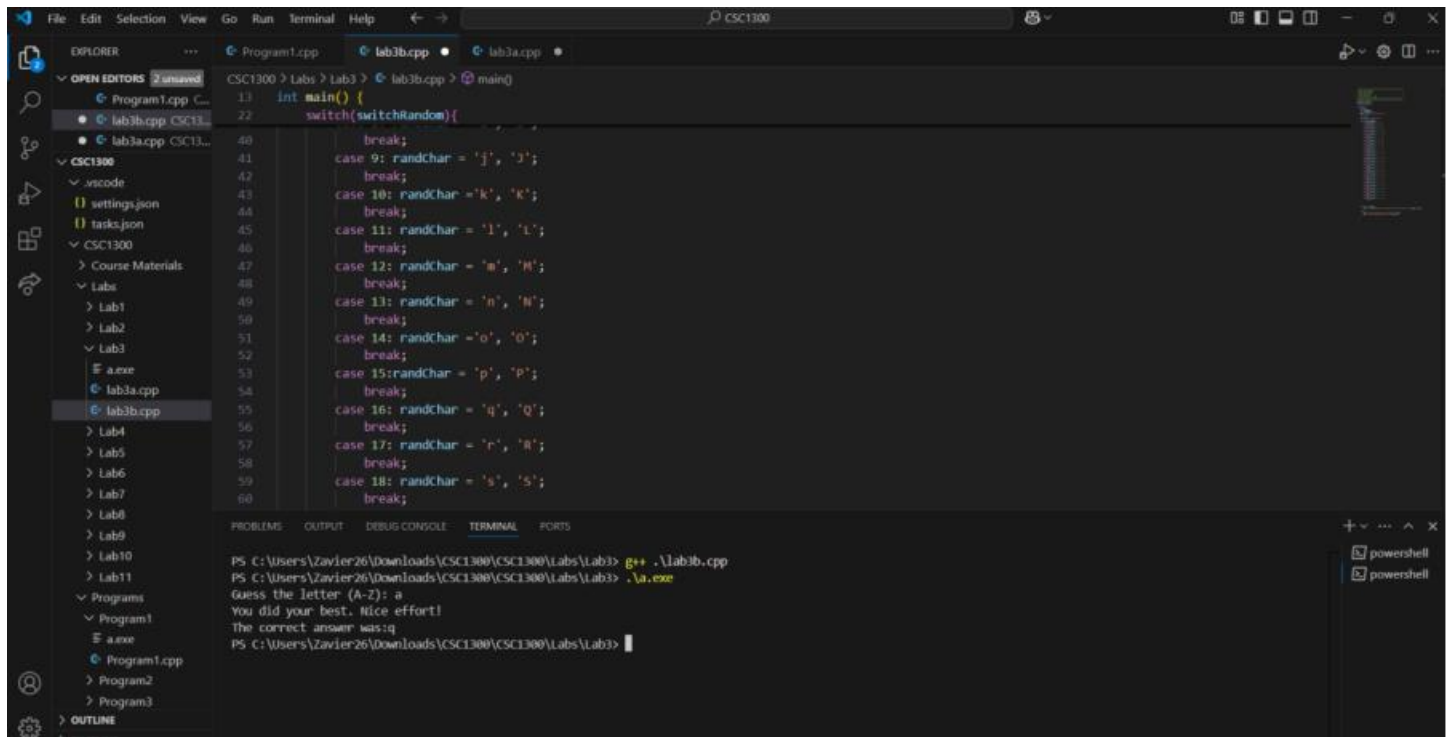
A screenshot of the Visual Studio Code editor interface. The Explorer pane on the left shows a project structure with folders 'CSC1300' and 'Labs', and files 'lab3a.cpp' and 'lab3b.cpp'. The main editor window displays the code for 'lab3b.cpp', which includes a switch statement for a guessing game. The Output pane at the bottom shows the command 'g++ .\lab3b.cpp' and the resulting output: 'Guess the letter (A-Z): a', 'You did your best. Nice effort!', and 'The correct answer was: q'. The terminal pane at the bottom right shows the command 'PS C:\Users\Zavier26\Downloads\CSC1300\CSC1300\Labs\Lab3> .\a.exe'.

STEP 3: FIX SYNTAX ERRORS

For each syntax error:

1. **Locate the error:**
 - a. Read the error message and identify the line number.
 - b. Look for red squiggles or warnings in your code editor.
2. **Understand the error:**
 - a. Hover over the error message or search online to understand what it means.
 - b. Document the meaning of the error here: When I finished typing out the rough draft of my Code there was no Syntax error to be found.
3. **Error Explanation:** N.A.
4. **Fix the error:**
 - a. Implement a solution and recompile the program.
 - b. Record whether the error went away.
5. **Repeat until all syntax errors are resolved**, adding a new Error Explanation and screenshot for each error you fix.

Screenshot 6:



The screenshot shows the Visual Studio Code interface. The Explorer panel on the left shows a project structure with folders like 'CSC1300' and 'Labs'. The main editor displays a C++ file named 'lab3b.cpp' with a switch statement. The code is as follows:

```
13 int main() {
22     switch(switchRandom){
40         break;
41         case 9: randChar = 'j', 'J';
42         break;
43         case 10: randChar = 'k', 'K';
44         break;
45         case 11: randChar = 'l', 'L';
46         break;
47         case 12: randChar = 'm', 'M';
48         break;
49         case 13: randChar = 'n', 'N';
50         break;
51         case 14: randChar = 'o', 'O';
52         break;
53         case 15: randChar = 'p', 'P';
54         break;
55         case 16: randChar = 'q', 'Q';
56         break;
57         case 17: randChar = 'r', 'R';
58         break;
59         case 18: randChar = 's', 'S';
60         break;
```

The bottom panel shows the 'TERMINAL' output:

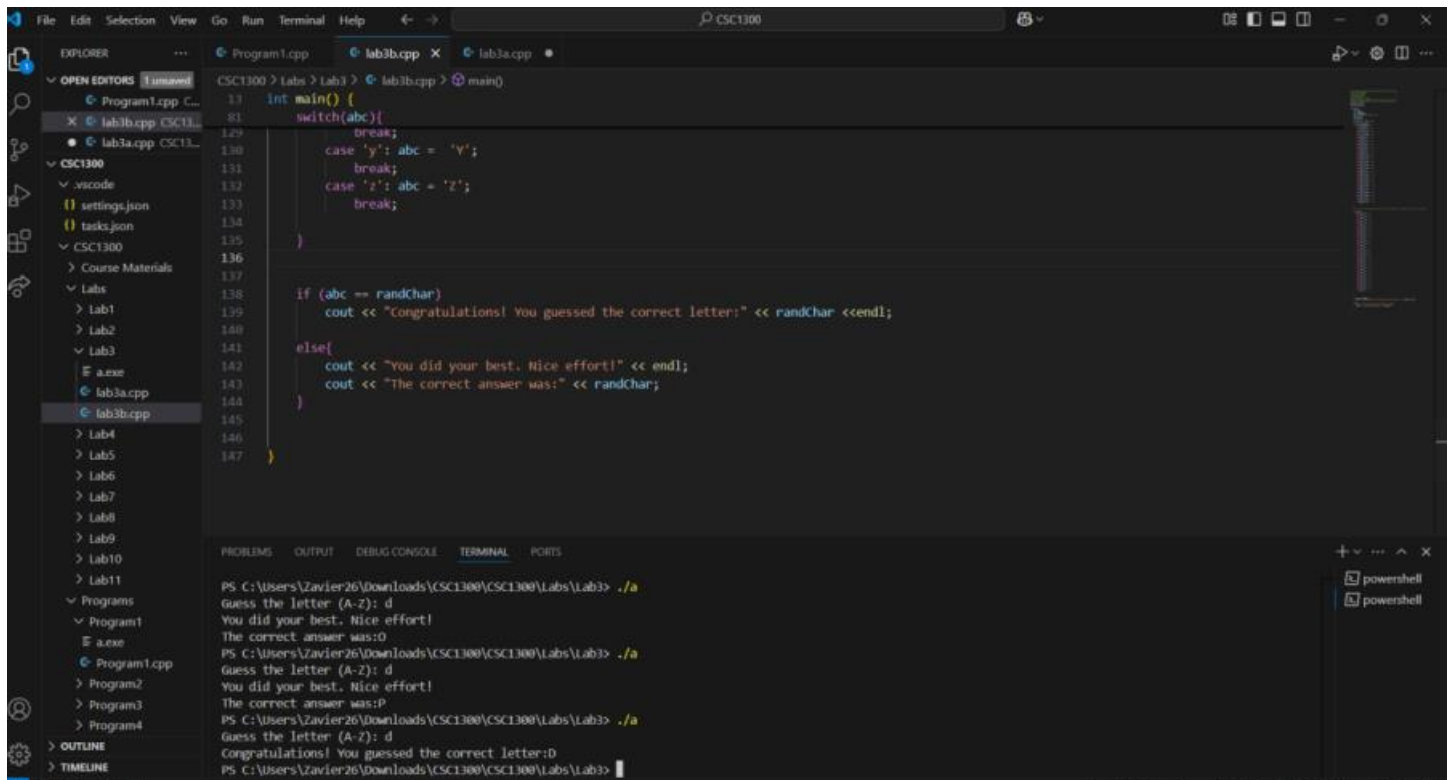
```
PS C:\Users\Zavler26\Downloads\CSC1300\CSC1300\Labs\Lab3> g++ .\lab3b.cpp
PS C:\Users\Zavler26\Downloads\CSC1300\CSC1300\Labs\Lab3> .\a.exe
Guess the letter (A-Z): a
You did your best. Nice effort!
The correct answer was:q
PS C:\Users\Zavler26\Downloads\CSC1300\CSC1300\Labs\Lab3>
```

STEP 4: ADDRESS LOGIC ERRORS

If the program runs but the output is incorrect or unexpected:

1. **Identify the problem:**
 - a. Compare the program's actual output to the expected output.
 - b. Describe what the program is doing versus what it should be doing.
2. **Observed Behavior:** The is giving a letter but it is not making my out have capital letters, It uses the same letter each time upon activation
3. **Expected Behavior:** The out put should have a capital letter not a lowercase. It should also use a different letter each time
4. **Test individual sections:**
 - a. Use print/cout statements or a debugger to isolate which part of the code is not working as intended.
5. **Locate and fix the error:**
 - a. Analyze the problematic section.
 - b. Document the issue and your solution:
6. **Logic Error:** The system would only recognize lower case letters rather than both
7. **Solution:** I created a switch statement that would convert the users input to Uppercase
8. **Re-run the program:**
 - a. Test whether your fix resolved the issue.
 - b. If not, repeat this step. **Do not** delete your previous responses or screenshot, just add to them with the new bug and new solution.

Screenshot 7:



The screenshot shows the Visual Studio Code editor with the file explorer on the left, the main editor window in the center, and the terminal at the bottom. The file explorer shows a project structure with folders for 'CSC1300', 'Lab1', 'Lab2', 'Lab3', 'Lab4', 'Lab5', 'Lab6', 'Lab7', 'Lab8', 'Lab9', 'Lab10', 'Lab11', 'Programs', and 'Program1'. The main editor window displays the code for 'lab3b.cpp' in the 'main()' function. The code includes a switch statement for handling user input and a conditional statement for providing feedback. The terminal at the bottom shows the output of the program, which is a series of prompts and responses for guessing a letter.

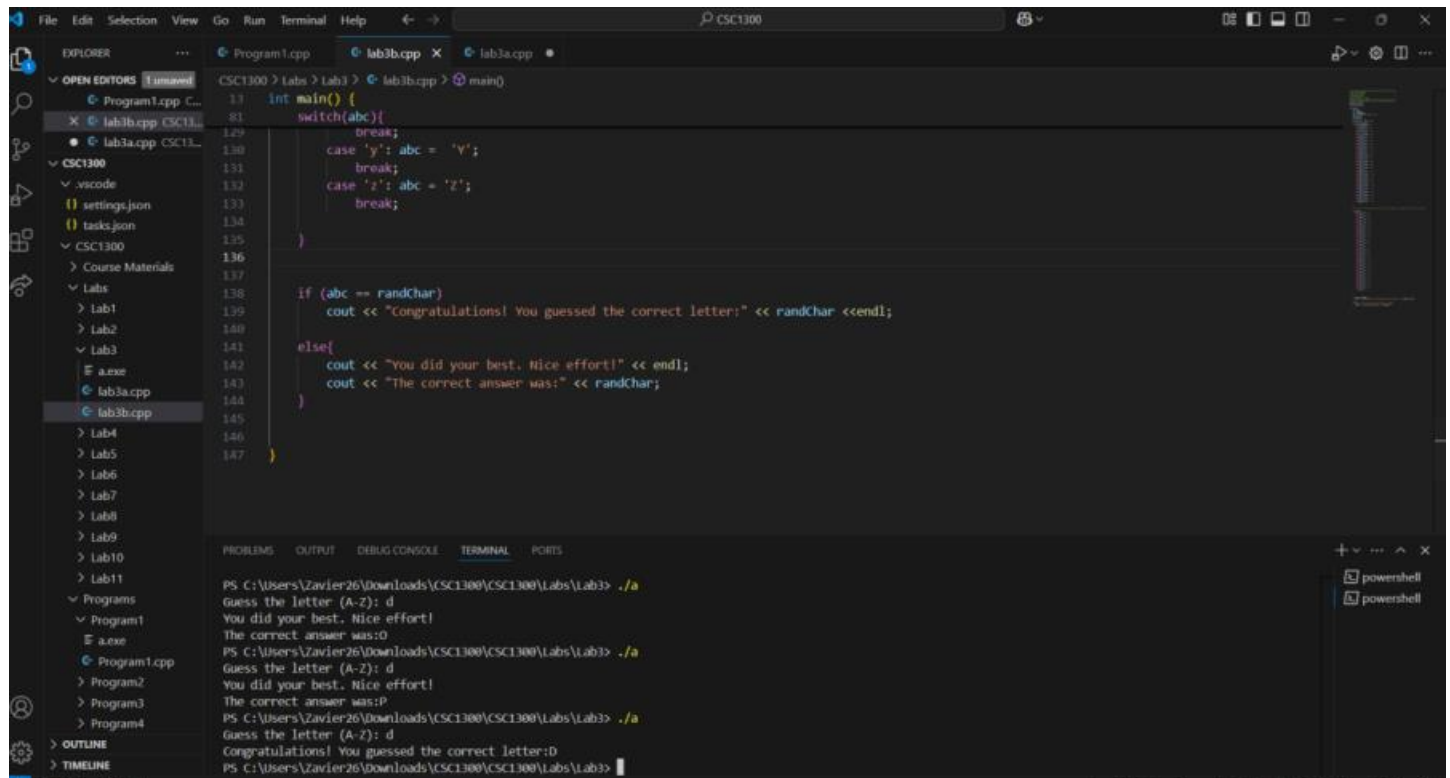
```
13 int main() {  
14     switch(abc){  
15         break;  
16         case 'y': abc = 'Y';  
17         break;  
18         case 'z': abc = 'Z';  
19         break;  
20     }  
21  
22     if (abc == randChar)  
23         cout << "Congratulations! You guessed the correct letter!" << randChar << endl;  
24  
25     else{  
26         cout << "You did your best. Nice effort!" << endl;  
27         cout << "The correct answer was:" << randChar;  
28     }  
29 }
```

```
PS C:\Users\Zavier26\Downloads\CSC1300\CSC1300\Lab3> ./a  
Guess the letter (A-Z): d  
You did your best. Nice effort!  
The correct answer was:0  
PS C:\Users\Zavier26\Downloads\CSC1300\CSC1300\Lab3> ./a  
Guess the letter (A-Z): d  
You did your best. Nice effort!  
The correct answer was:P  
PS C:\Users\Zavier26\Downloads\CSC1300\CSC1300\Lab3> ./a  
Guess the letter (A-Z): d  
Congratulations! You guessed the correct letter:D  
PS C:\Users\Zavier26\Downloads\CSC1300\CSC1300\Lab3>
```

STEP 5: ITERATE UNTIL CORRECT

1. **Continue debugging:**
 - a. Repeat each debugging process until the program runs correctly and produces the desired output.
2. **Document additional issues and fixes:**
 - a. Use the template from Steps 3 and 4 to track any new errors introduced during debugging.
3. **Final Output:**
 - a. Paste a screenshot of the final, correct output below:

Screenshot 8:



The screenshot shows the Visual Studio Code interface with the following components:

- EXPLORER:** Displays the file structure of the project, including 'Program1.cpp', 'lab3b.cpp', 'lab3a.cpp', and various configuration files like 'settings.json' and 'tasks.json'.
- EDITOR:** Shows the source code of 'lab3b.cpp'. The code includes a switch statement for handling guesses 'y' and 'z', and an if-else block for providing feedback based on the guess.
- TERMINAL:** Displays the output of the program execution. It shows the prompt 'Guess the letter (A-Z): d' followed by the program's response: 'You did your best. Nice effort! The correct answer was:0'.

```
int main() {
    switch(abc){
        break;
        case 'y': abc = 'Y';
        break;
        case 'z': abc = 'Z';
        break;
    }

    if (abc == randChar)
        cout << "Congratulations! You guessed the correct letter!" << randChar << endl;

    else{
        cout << "You did your best. Nice effort!" << endl;
        cout << "The correct answer was:" << randChar;
    }
}
```

```
PS C:\Users\Zavier26\Downloads\CSC1300\CSC1300\Labs\Lab3> ./a
Guess the letter (A-Z): d
You did your best. Nice effort!
The correct answer was:0
PS C:\Users\Zavier26\Downloads\CSC1300\CSC1300\Labs\Lab3> ./a
Guess the letter (A-Z): d
You did your best. Nice effort!
The correct answer was:P
PS C:\Users\Zavier26\Downloads\CSC1300\CSC1300\Labs\Lab3> ./a
Guess the letter (A-Z): d
Congratulations! You guessed the correct letter:D
PS C:\Users\Zavier26\Downloads\CSC1300\CSC1300\Labs\Lab3>
```

Note: Don't forget to go back to your **Spring 2025 Lab 3 Assignment** document to complete the final **Part 3** of the assignment and to see submission instructions.