CSCI1300 - Introduction to Computer Programming

Instructor: Dr. Knox

Due by the end of the recitation.

Recitation 3: Conditionals and Debugging

Conditionals

The if keyword is used to execute a statement or block, if, and only if, a condition is fulfilled. Its syntax is:

```
if (condition)
{
   // Execute these statements if condition TRUE
}
```

Sometimes, when the condition in an if statement evaluates to false, it would be nice to execute some code instead of the code executed when the statement evaluates to true. Use else if to specify a new condition to test, if the first condition is false. The "else" statement effectively says that whatever code after it is executed if the previous if statements are FALSE.
Its syntax is:

```
if ( condition 1 )
{
   // Execute these statements if condition 1 is TRUE
}
else if ( condition 2 )
{
   // Execute these statements if condition 1 is FALSE and condition 2 is TRUE
}
else
{
   // Execute these statements if FALSE
}
```

**REMEMBER: If an "if", "else if" or "else" condition is satisfied and entered, then the code will resume outside of the conditionals.*

Some common operators for comparison:

>	greater than	5 > 4 is TRUE
<	less than	4 < 5 is TRUE
>=	greater than or equal	4 >= 4 is TRUE
<=	less than or equal	3 <= 4 is TRUE
==	equal to	5 == 5 is TRUE
!=	not equal to	5 != 4 is TRUE

Some example code:

```
#include <iostream>

using namespace std;

void numberRange(int number)
{
    if(number > 0 && number <= 100)
    {
        cout << "This number is between 0 and 100!";
    }
    else if(number > 100 && number <= 200)
    {
        cout << "This number is between 101 and 200!";
    }
    else
    {
        cout << "This number is greater than 200!";
    }
}

int main()
{
    int number;
    cout << "enter a positive number to see what number range it falls in!\n";
    cin >> number;
    numberRange(number);
}
```

In today's recitation, you will need to complete two tasks: the first will be a debugging activity and the second will be a Moodle coding activity.

Debugging

Debugging and Working with Error Messages

For the remainder of this recitation, you will gain experience working through common syntax and coding mistakes and learning how to read error messages. When writing code, it is important to compile often so that you can quickly catch mistakes before too many accumulate! For this recitation you will be working with the file **recitation3primes.cpp.** In this file a number of errors have been built into the program intentionally, and it is your job to correct the syntax errors **AND** correct the code to print out the loops properly.

In the code, you will have 3 functions to debug. This program simulates many things that can go wrong with your code if you don't compile and debug systematically! **Don't forget, for this recitation you'll not only be fixing syntax errors, but you'll also be modifying the code to make the functions work properly!**

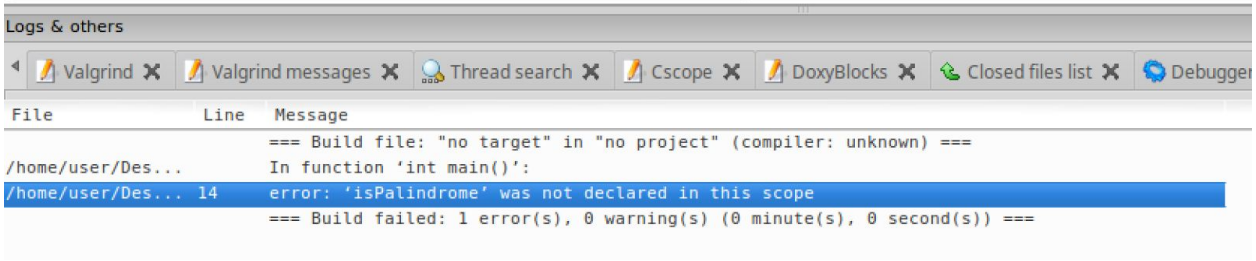
Getting Started

1. Download the **recitation3primes.cpp** file from Moodle and save it to one of your directories.
2. Open the file in **Code::Blocks** - File → Open... → **recitation3primes.cpp** → Open
3. Look over the code to get a sense of what the program is trying to do.
4. Compile(Build) and Run the code.

You will see a number of errors pop up on the Build Messages tab. The first error will be highlighted in red and this is the error that you should focus on correcting first. The way to read this error is to look at the line the error is occurring on or above it. Following is how it should look:

Error 1:

```
1  #include <iostream>
2  #include<bits/stdc++.h>
3
4  using namespace std;
5
6  int main()
7  {
8      string str = "";
9
10     cout << "Palindrome or not?" << endl;
11     cout << "Enter text: ";
12     cin >> str;
13
14     isPalindrome(str);
15 }
16
17
18 void isPalindrome(string str)
19 {
20     string originalstr = str;
```



In this case the code is breaking at **line 14.** The error message displayed on the highlighted line is

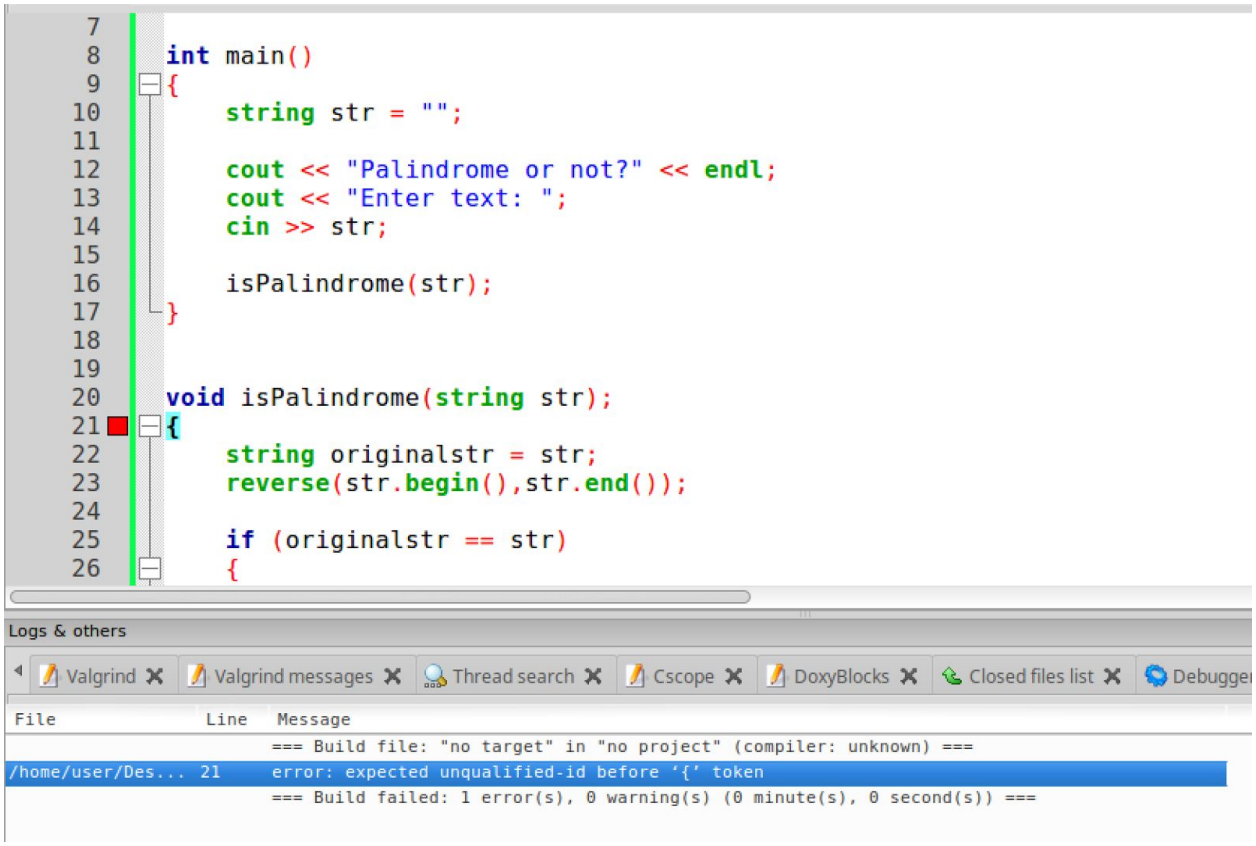
`error: 'isPalindrome' was not declared in this scope`

What is happening with this error? What line is it on?

This error is indicating that the isPalindrome function is not declared. What does this mean? Where should you declare your functions?

Working through this error means that you will have to initialize the function in the proper location so that your program recognizes isPalindrome as being within the scope!

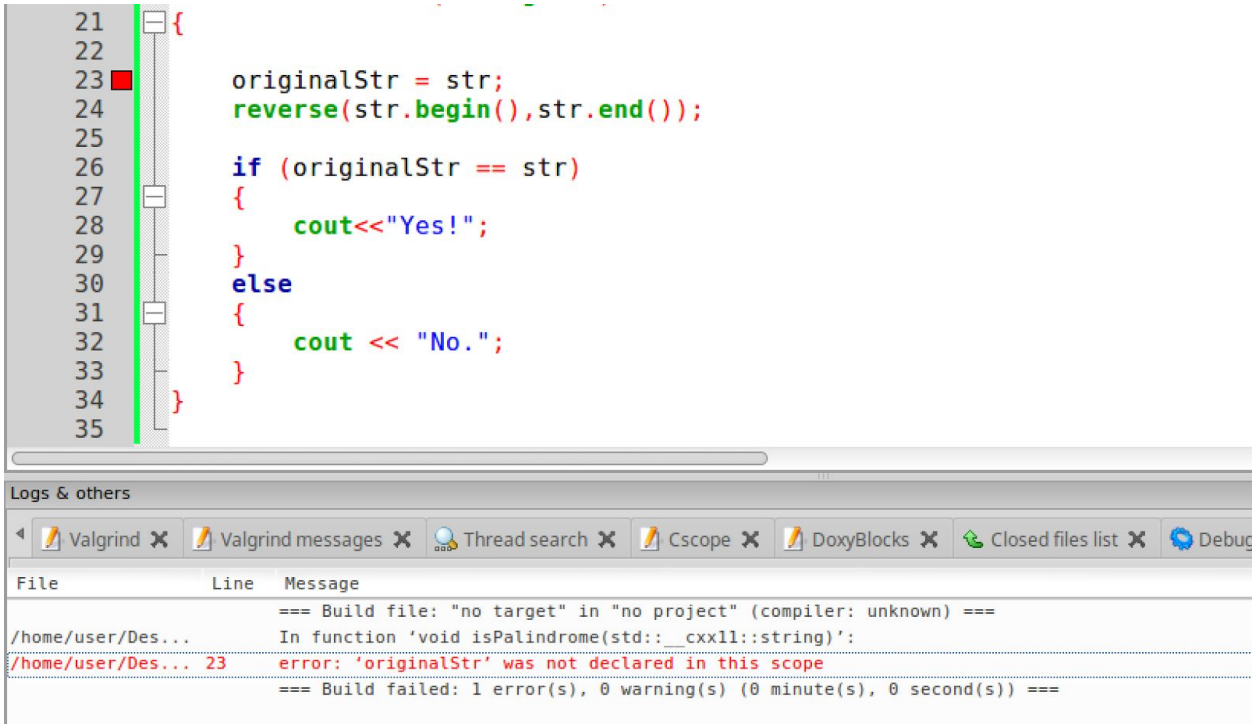
Error 2:




```
error: expected unqualified-id before ‘{‘ token
```

What does this error mean? Try googling the error to see why this has occurred. Take a look at **line 21**. This error indicates that a syntax problem is related to this line. Try looking at the line immediately before this error to find the problem.

Error 3:



error: 'originalStr' was not declared in this scope

What does this error mean? This error indicates that your program is expecting 'originalStr' to be declared prior to being used. What does this mean? Where should you declare your variables? Your variables should always be declared above the line of usage.

At this point you may encounter a few more errors that you need to continue to debug to make the program run.

Some Takeaways:

1. If you have a syntax error on a certain line sometimes the error actually occurred on the line BEFORE where your code is breaking.
2. Scoping errors require you include the proper libraries and namespaces in your source file. This kind of error lets you know to always look back and check which libraries you've imported.
3. Make sure to declare variables and functions prior to using and include the appropriate libraries.

Once you have finished debugging your program and completed all errors, verify that the program is running and its output is correct. Upload your **recitation3primes.cpp** file to Moodle at the **Recitation 3 - Debugging activity.**

.

Recitation Coding Activity

During every recitation we will go over what you need to know to prepare for your weekly homework. This will be achieved by completing the recitation quiz on Moodle.

Access the Recitation Programming Activity on Moodle
There are three problems for you to attempt and complete by the end of the recitation.

Note: *When you press the "check" at the bottom of the question, it will attempt to compile and run your version of the function with our test cases. The results will appear below your answer code.*

Some points to think about when you solve the questions:

- What code is only conditionally executed?
- Do you have to return a value, to print it out to the console, or both?
- Are there parameters? How many?
- Remember that we will provide the ***main*** function that will call your code. Only write the code for the functions specified in the question.

PA #1 - Modify the given code to implement a new functionality.
The first problem shows you a syntactically and semantically correct function declaration and code. Take inspiration from it to implement the code needed to achieve the new task required.

We need you to modify this code to change its behavior as listed in the question description.

PA #2 - You must write the code for the function declaration that we have provided. Inside the function definition implement the logic required to accomplish the task explained in the question.

PA #3 - You must derive the correct function declaration and write the code for the function code given a description. Write the complete function in the answer area.