## Lab Experience 1

Objectives:

1. Structure of the C++ program.
2. Compiling and correcting errors in a C++ program.
3. Entering and editing a C++ program.
4. Understanding Chapter 2 in the textbook.

When typing your program using the full screen editor all grammatical rules of the language must be strictly adhered to for the compiler to produce an executable program. There are two types of messages the compiler will generate: 1) A warning message and 2) A syntax error. A warning message is used by the compiler to tell you of a potential error, but it does not halt the compilation process. The programmer has the choice of fixing the warning message or ignoring it.

When working with dynamic memory allocation you should fix all warning messages.

A syntax error means the grammar of the language has been violated and program compilation will end. The programmer has no choice but to fix these types of errors since an executable will not be created until all syntax errors have been corrected.

When correcting syntax errors begin with the first one in the list since it is likely all others will have been generated by the first syntax error.

Another type of error not associated with grammar of the language is a **logic** error. This is caused by a faulty algorithm created by the programmer. These are harder to correct (debug) since you have to track down the error by “walking through” the algorithm. “Walking through” an algorithm means you have to be the computer and track all changes made by the program to discover the error.

This can be a time consuming task and because of this most experienced programmers will desk check the algorithm before translating the algorithm into source code. It is possible the algorithm developed by the programmer will not solve the problem and a new algorithm/approach will be needed. Another technique used by programmers is to compile a partially completed program to determine if everything is working correctly.

Whenever you make a change to your program you must recompile the program for the changes to take effect. If you don’t the old executable will always be used when you execute your program.

In this lab you will learn how to use C++ escape sequences, discover the differences between integer and floating point division, and learn how to correct the most the common syntax errors by creating the error and then fixing the error.

NOTE: DO NOT COPY AND PASTE THE CODE SINCE IT CONTAINS NON-ASCII CHARACTERS.

**What you must do:**

**Problem 1:**

1. Start .Net IDE to create a new C++ source file called lab1prob1.cpp.
2. Type in the program below.

// Program to print part of a song

#include <iostream>

using namespace std;

int main(void){

cout << "Give me land, lots of land" << endl;

cout << "And the starry skies above..." << endl;

return 0;

}

At this point, the program is listed below with line numbers for you to refer to for the completion of problem 1.

* + 1. // Program to print part of a song
    2. #include <iostream>
    3. using namespace std;
    4. int main(void)
    5. {
    6. cout << "Give me land, lots of land" << endl;
    7. cout << "And the starry skies above..." << endl;
    8. return 0;
    9. }

1. Given the program above, answer the questions below in your word document called lab1.doc. Copy and paste the questions into your document and answer the questions.  **Place your name and lab section number in the upper left hand corner of your document.**
   1. Line 1 is called a \_\_\_**Header Comment**
   2. Line 2 is a \_\_\_\_\_\_\_\_**Preprocessor Directive**
   3. What is the purpose of line number 2? **To tell the computer to include the IO stream data**
   4. What is the purpose of line #4? **It begins the function of the program**
   5. Lines 6 and 7 utilize the symbols, <<, what is the name of this operator and how is it used? **The Insertion Operator. It inserts data into the IO stream**
   6. Replace **<< endl;** in statement 6 with a semicolon (**;**). Describe the behavior of your program with this change. **The text is all continuous; there is no line break.**
   7. Replace **land”;** in statement 6 with **land\n”;** Describe the behavior of your program with this change. **The line breaks after land; it looks identical as the << endl version.**

**Problem 2: Creating an Error Log**

// Program to print part of a song

#include <iostream>

using namespace std;

int main(void){

char letter;

letter = ‘A’;

cout << "Give me land, lots of land";

cout << "\nAnd the starry skies above...\n";

cout << letter << endl;

return 0;

}

1. Modify your program from **problem1, Lab1prob1.cpp**, to look exactly like the above program. After modifying the program, build and execute the program. If the program does not execute, correct all of your mistakes (any mistakes will be typographical).
2. In this problem you will be making errors on purpose. Make the following changes in your program described below and execute your program. Write the description of the error and what caused it in your word document **lab1.doc**. Correct the error and rerun the program to verify that the program executed correctly. If the compiler does not flag an error, so state. If the output produced does not look exactly like the output in step 1, then you have a logic error. If a change does not cause an error message, so state.

State what the logic error is and how you corrected it. **Note: Don’t just copy and paste the compiler messages into your word document as your final answer. This is good, but I want you to use your own words in describing the error and how you corrected it. This will enhance your learning of the material and assist you in debugging your own programs in the future.**

1. Delete // of the opening comment

**Error Message Generated:** A Build error came up

**Type of Error:** Syntax Error

**What caused the error and how I corrected it:** The error was caused by the the compiler reading the text as is; it was trying to interpret it as a statement to do work, as opposed to a comment. I added the two forward slashes (//) back to make it a comment again. It can be seen as text color green to confirm it is a comment in Visual Studio.

1. Comment out the preprocessor directive: #include <iostream>

**Error Message Generated:** A build error popup

**Type of Error:** Syntax Error

**What caused the error and how I corrected it:** A build error was caused by not including the iostream. This can be corrected by deleting the comment slashes.

1. Put a semicolon at the end of the preprocessor directive

**Error Message Generated:** None

**Type of Error:** None

**What caused the error and how I corrected it:** There was no error that I could see.

1. Remove the right curly brace from the program.

**Error Message Generated:** Build Error popup

**Type of Error:** Syntax Error

**What caused the error and how I corrected it:** The right curly brace was not at the end of the main. I added the right brace back in to make it run correctly.

1. Replace the << with < in the first cout statement.

**Error Message Generated:** No error message

**Type of Error:** Syntax Error; Text displayed into console did not display "Give me land..." string

**What caused the error and how I corrected it:** By taking the second < out, it seems to skip over that line since it did not read it as the insertion operator. To correct it, I added the second < back in.

1. Remove the apostrophes from the character constant ‘A’.

**Error Message Generated:** Build Error popup

**Type of Error:** Syntax Error

**What caused the error and how I corrected it:** The constant could not be identified because the apostrophes were not there for the computer to identify the A as a character. To correct it, I added the apostrophes back around the character A

1. Change the apostrophes in ‘A’ to “”

**Error Message Generated:** Build error popup

**Type of Error:** Syntax error

**What caused the error and how I corrected it:** The computer could not identify the character A with the quotes around it. To correct, I replaced the quotes with apostrophes.

1. Reverse the positions of the statements char letter; and letter = ‘A’;

**Error Message Generated:** Build Error

**Type of Error:** Logic Error

**What caused the error and how I corrected it:** The declaration of the character "letter" came after the assignment of the "letter" to A; You cannot assign something that has not been declared because the memory has not been allocated yet. To correct, I switched the statements back.

1. Remove the \ from \n in the second cout statement.

**Error Message Generated: None**

**Type of Error:** None; however the text did not display out as intended to by the original text.

**What caused the error and how I corrected it:** Error was caused by the lack of the escape sequence as marked by the \. To correct, the \ was added back.

1. Place \ in front of both \n’s

**Error Message Generated:** None

**Type of Error:** None; however the text did not display out as intended to by the original text.

**What caused the error and how I corrected it:** the additional \ caused the computer to print the \n escape sequence as is, instead of treating it as an escape sequence. This was corrected by removing the additional \'s

1. Change the statement:

cout << "\nAnd the starry skies above...\n";

to

cout << "\nAnd the \”starry\” skies above...\n";

Did the output change? If yes, what was the change?

Yes, the starry now had quotes around it like "starry"

**Error Message Generated:** none

**Type of Error:** None; however the text did not display out as intended to by the original text.

**What caused the error and how I corrected it:** the \" caused a quote character to print. To correct, I removed the \"

1. Remove the statement return 0; from the program.

**Error Message Generated:** None

**Type of Error:** None

**What caused the error and how I corrected it:** No error was generated.

**Due Dates:** According to the due date posted for the drop box folder.

**What to hand in:**

1. Place the word document into the lab1 drop box. Save the word document as: yourNameLab1.doc or docx.
2. Hand in a print out of your word document.