

Laboratory Exercise 1 (C++ Review)

Topics: C++ Review

Compiling and executing C++ programs on a Linux platform

C++ strings

file input

console display output

loops

decision statements

simple arithmetic

cin.fail() condition test

using a debugger

Goals: Upon successful completion of this lab you should be able to:

1. Start a linux session
2. Enter and edit a C++ program
3. Compile and run a C++ program
4. Rename and submit a C++ program to the drop box
5. Complete a linux session
6. Write a C++ input statement
7. Write a C++ output statement
8. Predict when the C++ cin.fail () test will return 1
9. Use the gdb debugger

Related text sections:

Chapter 2 (C++ program format)

emacs reference

linux reference

Laboratory Exercise 1 Pre-lab questions

Note: you can use your book or the web to answer the following questions. cplusplus.com is a very useful website.

1. What are `cin` and `cout`?

2. What statements are used to open a text file as input to a program?
 - a. Includes?

 - b. Variable declarations?

 - c. Open statement?

3. How do you create an input file for a program?

4. What do the functions `isalpha` and `isdigit` and `isspace` do?

5. What do the functions `tolower` and `toupper` do?

6. Given the following C++ statements

```
char c;  
c = 'a';
```

how do the statements

```
toupper (c);
```

and

```
c = toupper (c);
```

differ?

Laboratory Exercise 1 Instructions

Part 1

1. Boot the computer and select the Linux operating system.
2. Logon as “student”
3. Start a console session.
4. Use ssh to remotely connect to the mainframe:
`ssh username@ewolf.cs.sonoma.edu`

Note: steps 1 through 4 can also be done using the Windows operating system and TeraTerm SSH or Putty.

5. You may need to change your password:
`passwd`
6. Create and switch to a CS 215 subdirectory and a Lab1 directory within your CS 215 directory:
`mkdir cs215`
`cd cs215`
`mkdir Lab1`
`cd Lab1`
7. Using emacs, vi, or another text editor, enter the following C++ program (call the file Lab1a.cpp):

```
#include <iostream>
using namespace std;

int main ()
{
    int v;
    cout << "Enter the data set input: ";
    cin >> v;
    cout << v << (cin.fail() ? " true" : " false") << endl;
    cin >> v;
    cout << v << (cin.fail() ? " true" : " false") << endl;
    cin >> v;
    cout << v << (cin.fail() ? " true" : " false") << endl;
    return 0;
}
```

8. Compile the program using the g++ compiler (call the executabel lab1a):
`g++ Lab1a.cpp -o lab1a`

9. Execute the program several times:
`./lab1a`

Use each of the data sets in the following table as input. Each time the program asks you to enter the data set, enter all of the characters in the data set the first time the program pauses for input. Record the output in the table on the next page.

9. (continued)

Case	Input data	Observed Output
1.	5 12 15	
2.	5 -12 15	
3.	5 a 12	
4.	5.13 -12 15	
5.	abc 5 12	

10. Write 1 to 3 sentences describing your observations of the rules used by the input operator (>>) when it is used to read an integer value.

11. Modify the program by changing the type of v from int to float.

12. Recompile and run the program with each of the data sets in the following table (Enter all of the input in the data set the first time the program pauses for input). Record the output.

Case	Input data	Observed Output
1.	5.1 12.05 15.23	
2.	5.1 -12.05 15.23	
3.	5 12 15.23	
4.	5.13 a 15	
5.	abc 5 12	

13. Write 1 to 3 sentences describing your observations of the rules used by the input operator (>>) when it is used to read a float value.

14. Modify the program by changing the type of v from float to char.

15. Recompile and run the program with each of the data sets in the following table (Enter all of the input in the data set the first time the program pauses for input). Record the output.

Case	Input data	Observed Output
1.	a b c	
2.	a b c	
3.	abc	
4.	5 a 12.5	
5.	abc 5 12	

16. Write 1 to 3 sentences describing your observations of the rules used by the input operator(>>) when it is used to read a character value.

17. Modify the program by changing the type of v from char to string. Note: you will also need to add another include directive: `#include <string>`

18. Recompile and run the program with each of the data sets in the following table (Enter all of the input in the data set the first time the program pauses for input). Record the output in the table on the next page.

18. (continued)

Case	Input data	Observed Output
1.	abc def ghi	

2.	abc def ghi	
3.	a b c d e f g h	
4.	5.13 -12 15	
5.	abc 5 12	

19. Write 1 to 3 sentences describing your observations of the rules used by the input operator(>>) when it is used to read a C++ string value.

20. Using a text editor, create a document called Lab1a.txt that describes the rules for reading integers, floats, characters, and C++ strings from an input stream. Each of your rules should describe exactly what will be read from the input stream and when the stream will fail. Submit your Lab1a.txt file as yourlastnameL1.txt.

```
cp Lab1a.txt yourlastnameL1.txt
cp yourlastnameL1.txt ~tiawatts/cs215drop/.
```

Part 2

1. Write a program called Lab1c.cpp based on the following specifications:
 - A. Your program should open, for input, a text file called “words.txt”. This file will contain several lines of text.
 - B. Your program should count the number of words in the file. A word is defined as a string of characters delimited by white space.
 - C. Your program should count the number of vowels in the file. Only A(a), E(e), I(I), O(o), and U(u) should be counted as vowels by your program.
 - D. Your program should count the number of consonants in the file.
 - E. Your program should count the number of digits in the file.
 - F. Your program should count the number of special characters in the file. Any non-white-space character that is not included in the counts described in statements C, D, or E should be counted as special characters by your program.
 - G. Your program should print out a summary of the counts.

H. Sample Input and Output:

words.txt

```
This file contains 15 words.  
It has lots of letters and  
very few special  
characters!
```

console output

```
Words: 15  
Vowels: 22  
Consonants: 44  
Digits: 2  
Special characters: 2
```

2. Compile your program using the g++ compiler; create an executable call lab1c:
3. Using a text editor, create a file called words.txt for testing your program; you can use the sample data shown above.
4. Run the executable:

```
./lab1c
```
5. If your program does not execute correctly, modify it, recompile it and execute it again.
6. When you are sure that your program is executing correctly, make a copy of the program to put in the drop box and move your copy to the drop box:

```
cp Lab1c.cpp yourlastnameL1.cpp  
cp yourlastnameL1.cpp ~tiawatts/cs215drop/.
```
7. Wait about 2 minutes and then check the web page to verify that you file has reached the drop box:
<http://www.cs.sonoma.edu/~tiawatts/cs215/lab1sub.txt>

1. What is the output of the following program

```
#include <iostream>
#include <string>
using namespace std;

int main ()
{
    int i = 0;
    float f = 0.0;
    char c = '?';
    string s = "???" ;

    cout << "Enter line of data: ";
    cin >> i >> f >> c >> s;

    cout << "The output is: ";
    cout << i << ' ' << f << ' ' ;
    cout << c << ' ' << s << endl;

    return 0;
}
```

- a. When the input is:

12 13.45 a bcdef

- b. When the input is:

13.45 a bcdef

- c. When the input is:

fedcb a 54.31 21

- d. When the input is:

1213.45abcdef