LAB 2

Lab Objectives:

- See common errors when writing C++ programs and how to fix them.
- Practice good coding standards and follow the CS1570 Coding standard.
- Debug the code given to you.
- Submit the debugged code using the 'cssubmit' tool.

Common Errors made my beginners:

lab2.cpp: No such file or directory
You misspelled the name of the file you wanted to compile, or left off the .cpp extension when you typed the file in.

Parse error at the end of input
Parse error before '}'
These errors usually occur because braces are not matched up somewhere. While the error will usually specify the last line of the file, the actual omitted brace is probably somewhere else in your program. These are often difficult errors to find!

"cin" undeclared
"cout" undeclared
These errors occur when you forget (or misspell) the #include <iostream> statement at the beginning of your program. Note that there is NO space between # and include.

No match for ‘istream & << int &’
No match for ‘istream & >> ostream &(&) (ostream &)’
These errors occur when the wrong input/output operator is used in cin or cout statement.
(Remember: use >> for input (cin’s), << for output (cout’s)).

error: expected ';' before…..
This happens when a semi-colon (‘;’) is missing at the end of a line in the code.
Ex:
sample.cpp:8:1: error: expected ';' before '}' token
In the above error message, somewhere before the line 8 in the code, a semicolon is missing.
Note that most of the times the compiler will give you the line number where it encountered the error.

Missing terminating “character
This error occurs when the closing quotes
on a string are left off, as in
cout << “This string is not closed << endl;

Redeclaration of ‘int var’
The variable var has been declared more than once. Sometimes this happens if you try to use the same variable name for two different purposes.

Non-lvalue in assignment
The left-hand side (called lvalue) of an assignment statement must be a single variable (as in a=b+4), not a constant (as in 3=a). While in

algebra the equals sign means “is equal to”, in C++ it means “is replaced by” and the order cannot be interchanged.

‘var’ undeclared

The variable name listed was not declared (with an int or float statement), or was misspelled either here or in the declaration.

Note:

1) Spelling mistakes:

   Please be patient and pay attention while you enter the code.

   `iostream ≠ isostream`

   `endl ≠ end1`

   `cout ≠ cot`

2) C++ is case sensitive which means

   `Endl ≠ endl`

Tip:

When the compiler gives a list of errors, don’t get frightened, just concentrate on the first few lines of the list to know where exactly the error is. Also, the compiler gives you the line number of the error, try to locate the error with the help of the line number.

Coding Standard

The coding standard is an 11-page document listed on Dr. Price's website. Eventually, you'll be expected to follow the entire coding standard. For now however, make sure your lab meets the following requirements:

- The comment section at the top of your lab is filled out and all the appropriate sections are listed (see below).
- The contents of the main function are indented 2 spaces.
- Curly braces { and } go on their own lines.

i) Top Section

A correctly formatted top section should look similar to this:

```
// Programmer: <Put Your Name>     Date: <Today's date>
// File: Lab2.cpp                Class: CS1580 Section E
// Purpose: Description of the function of the program.
```

ii) Indentation and Curly Braces

Anything between a pair of curly braces { and } should be indented 2 spaces. Each time you open a curly brace, indent over two more spaces. Do not use tabs—these print differently than they are viewed, and will result in line wraps!

Additionally, curly braces should always go on their own line.

```
int main()
{
  // I'm inside some curly braces, so I've indented two
spaces!
  cout<<"I'm the coolest kid on my block!"<<endl;
  return 0;
}
```

Lab Preparation:

- Download lab2.cpp from canvas and place it on your desktop
- Using Putty, connect to the CS linux machine and enter your login credentials.
- Go to SDRIVE (cd SDRIVE)
- Go to the directory cs1580 (cd cs1580)
- Create a directory named lab2 (mkdir lab2)
- Move lab2.cpp from the directory Desktop to the directory lab2. (mv ../Desktop/lab2.cpp lab2/) . Here, to move files from one directory to another we use the "mv <source> <destination>" command.
- Go to the directory Lab2 (cd lab2)
- Using JPICO editor open lab2.cpp and start debugging. (jpico lab2.cpp)
- To compile,
  fg++ lab2.cpp –o <enter any name>
- To submit,
  cssubmit 1580 <section> <assignment number>

To get 100 points in this lab assignment:

- Enter your info properly in the header of the code.
- Compile successfully.
- Meet the portion of the coding standard listed in this lab.