

Author: Tatyana Vlaskin
Assignment 1 Report

Understanding:

First part of this assignment was straight forward. I had to write short summary about myself, paste my picture in the document and save it as a pdf file.

Second part of this assignment was to become familiar with the number of tools that are essential for getting a passing grade in this class. It is allowed to use any IDEs to write a code. However, in order to make sure that the source code compiles on the OSU server (FLIP), it is recommended to write source code in the text editor (VIM is one example). After source code is written, we need a way to transfer source code on the FLIP server and a way to connect to the FLIP server. This can be accomplished using FTP client program (FileZilla) and secure shell program (SSH) (PuTTY is one example). Once we are connected to the FLIP server, we use a terminal to communicate with the server. To compile our source code (translate it into the machine readable language) we use g++. You have to be in the right directory in order to compile a file. Otherwise, you'll get a message: "No such file or directory." To fulfill this part of the assignment, I had to create ENGR account, download VIM, FileZilla and PuTTY. (It is also possible to create source file on the server, but I decided not to take that route).

After all the programs were downloaded, I had to make sure that everything works and that I have some basic knowledge of the server and how to navigate the file-systems on the server. I was exposed to such commands as: ls, cd, rm, mv. I've also created new directory (CS161) on the server, where I am planning to keep all my work.

The final part of the Assignment was to write two short programs: hello and echo.

Hello program needs to print a message to the console: Hello World!

Echo program needs to print a message to the console asking a user to enter an integer, once the integer is entered, the program needs to print a second message to the console as well as to include the integer that was entered by the user.

Design:

For the first part of assignment I had to decide what I want to write about myself, find a picture and figure out how to save file in a pdf format.

For the second part of the assignment, I went to

https://secure.engr.oregonstate.edu:8000/teach.php?type=want_auth to set up my ENGR account. Next I downloaded VIM, FileZilla and PuTTY and made a stepwise instruction how to test source code on the FLIP. To make sure that all the programs work, I decided to compile

one of the sample codes provided on the class website. This guaranteed that if something does not work the issue is not because of the bug in the code.

INSTRUCTIONS TO TEST PROGRAM ON FLIPP:

SAVING A SOURCE FILE:

1. Open VIM
2. Press I to go enter insert mode
3. Copy and paste code from the website.
4. Save file as hello.cpp
5. Type :w RETURN
6. Type :q RETURN to exit VIM

TRANSFERRE SOURCE CODE

1. Open FileZilla
2. Connect to: access.engr.oregonstate.edu host. Make sure that port 22 is selected.
3. Make new directory CS161 on the FLIP server.
4. Copy source hello.cpp file to the FLIP

CONNECT TO THE FLIPP SERVER:

1. Open PuTTY
2. Log in the FIPP server: (vlaskint@access.engr.oregonstate.edu)
3. Type cd to make sure that you are in the home directory
4. Type CS161 to open that directory
5. Compile file using: g++ hello.cpp -o hello
6. Run file typing ./hello

Once everything worked I started thinking about the source code for hello and echo programs.

hello program is a simple one string command.

The program will be started with the following two lines that set up things in such a way that the library with the console output facilities are available for the program.

```
#include <iostream>    // Macro that includes library
```

```
using namespace std;  //Global statement
```

the program will end with return (0); to indicate to the computer that that is the end of the program.

echo program will be started with the same 2 first lines as hello program:

```
#include <iostream>    // Macro that includes library
```

```
using namespace std;  //Global statement
```

Next step will be a variable declaration. I decided to use int type. Actually, I could have used short type as well because it has enough memory to store number as high as 32,767 and

people have a tendency not to use numbers over 10K. However, I decided to use int type because it seems like a commonly used type for integers.

After variable declaration the program needs output operator that will print message to the console for use to enter an integer. In C++ output operator is cout <<.

Next step is the cin and >> operators to read the input from user.

One more cout << is needed to display the integer back to the user.

Finally the program will end with return (0); to indicate to the computer that that is the end of the program.

Tests Done:

1. To make sure that VIM, FileZila and PuTTY programs work, I decided to compile one of the sample codes provided on the class website. That guaranteed that if something does not work the issue is not mistake in the code.
2. To test hello program, I complied and ran it on the FLIP.
3. To test echo program, I complied and ran program on the FLIP. I checked what happens when integer, floating point number, word, character, and sentence is entered.

Reflection/Discussion:

After this assignment I am familiar with the number of tools/programs/systems that are essential in this class.

I've also learned how to use Standard input and Standard output operators.

I've learned the difference between different simple types in C++ and how they are used.

Primitive types: short int and long int are available to control how much memory will be allocated to the number. If variable is expected to be a small number, programmer can use short. For example if the declared variable will represents age, weight of human, number of teeth, etc than programmer can use short type. If number is expected to be large, programmer should use long. An example can be number of molecules in a container.

In C++ the insertion operator is cout << and extraction operator is cin >>. For these statements to work you need to use the global statement such as: using namespace std;. Alternative way is to designate the namespace for cout, cin and endl separately:

```
using std::cout;
```

```
using std::cin;
```

```
using std::endl;
```

I prefer to use global statement.

I think that if the user enters a word, sentence, character or number higher than 2,147,483,647, an error message will be displayed and program will stop before completion. If floating point number is entered, it will be truncated and only number before decimal point is displayed back.

When I tested my prediction, it turned out that I was wrong about word, sentence, character or number higher than 2,147,483,64. In reality "0" is displayed back.

I was right about floating point number.