Lab 1 Readme Doc

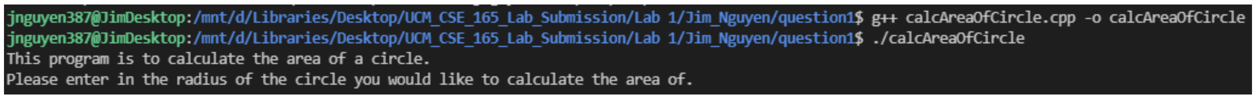
By: Jim Nguyen

Lab 02L

9/17/2021

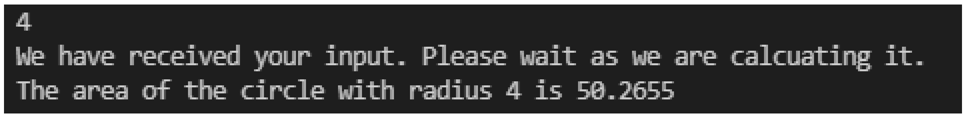
Question 1:

For question 1, I was tasked to code a program to find a circle's area given a user input for the radius. The image below shows how I compiled the program on my computer.



```
jnguyen387@JimDesktop:/mnt/d/Libraries/Desktop/UCM_CSE_165_Lab_Submission/Lab 1/Jim_Nguyen/question1$ g++ calcAreaOfCircle.cpp -o calcAreaOfCircle
jnguyen387@JimDesktop:/mnt/d/Libraries/Desktop/UCM_CSE_165_Lab_Submission/Lab 1/Jim_Nguyen/question1$ ./calcAreaOfCircle
This program is to calculate the area of a circle.
Please enter in the radius of the circle you would like to calculate the area of.
```

After that, it requires a user input of a single number. For the image below, I have selected the number 4. The answer will then be shown after a second or two.



```
4
We have received your input. Please wait as we are calcuating it.
The area of the circle with radius 4 is 50.2655
```

Then the program promptly ends. I have also left a copy of Stream2.cpp and Numconv.cpp for me to reference as I made the code.

Here are just a written-out versions of the images above.

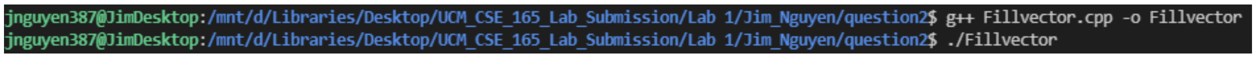
Compile: g++ calcAreaOfCircle.cpp -o calcAreaOfCircle

Run: ./calcAreaOfCircle

Input(cin): any integer value.

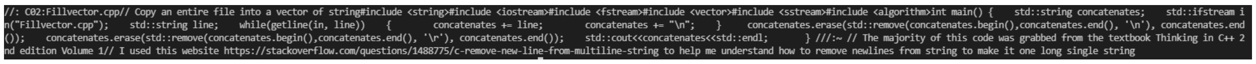
Question 2:

For question 2, I had to change the FillVector.cpp that was in the textbook so that it puts all the elements into a single string. The image below shows how I compiled the program on my computer.



```
jnguyen387@JimDesktop:/mnt/d/Libraries/Desktop/UCM_CSE_165_Lab_Submission/Lab 1/Jim_Nguyen/question2$ g++ Fillvector.cpp -o Fillvector
jnguyen387@JimDesktop:/mnt/d/Libraries/Desktop/UCM_CSE_165_Lab_Submission/Lab 1/Jim_Nguyen/question2$ ./Fillvector
```

After that, it will promptly show you the output of the single string. There is no user input necessary as it is using the file itself (Fillvector.cpp) as the input.



```
//: C02:Fillvector.cpp// Copy an entire file into a vector of string#include <string>#include <iostream>#include <fstream>#include <vector>#include <sstream>#include <algorithm>int main() {    std::string concatenates;    std::ifstream i
n("Fillvector.cpp");    std::string line;    while(getline(in, line))    {        concatenates += line;        concatenates += "\n";    }    concatenates.erase(std::remove(concatenates.begin(),concatenates.end(), '\n'), concatenates.end
());    concatenates.erase(std::remove(concatenates.begin(),concatenates.end(), '\r'), concatenates.end());    std::cout<<concatenates<<std::endl;       } ///:~ // The majority of this code was grabbed from the textbook Thinking in C++ 2
nd edition Volume 1// I used this website https://stackoverflow.com/questions/1488775/c-remove-new-line-from-multiline-string to help me understand how to remove newlines from string to make it one long single string
```

Then it promptly exits the program.

Here are just a written-out versions of the images above.

Compile: g++ Fillvector.cpp -o Fillvector

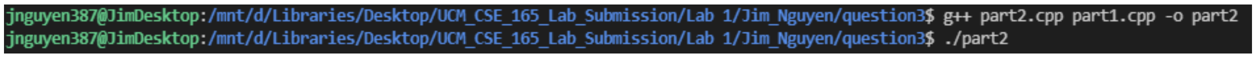
Run: ./Fillvector

Input(file): Fillvector.cpp (In question 2 folder)

Question 3:

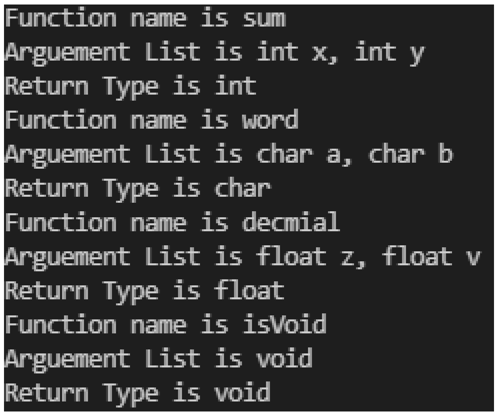
For question 3, it wanted me to create a header file and declare a group of functions, then it wants me to create a .cpp and create definitions for the functions. Then it wants me to create another .cpp file that includes the header file and run all the functions in it.

The image below shows you how I compiled my program.



```
jnguyen387@JimDesktop:/mnt/d/Libraries/Desktop/UCM_CSE_165_Lab_Submission/Lab 1/Jim_Nguyen/question3$ g++ part2.cpp part1.cpp -o part2
jnguyen387@JimDesktop:/mnt/d/Libraries/Desktop/UCM_CSE_165_Lab_Submission/Lab 1/Jim_Nguyen/question3$ ./part2
```

After that, it will output what the function prints out. There is no user input or file input for this program.



```
Function name is sum
Arguement List is int x, int y
Return Type is int
Function name is word
Arguement List is char a, char b
Return Type is char
Function name is decmial
Arguement List is float z, float v
Return Type is float
Function name is isVoid
Arguement List is void
Return Type is void
```

Here are just a written-out versions of the images above.

Compile: g++ part2.cpp part1.cpp -o part2

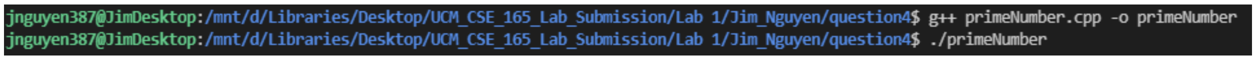
Run: ./part2

Input: no input necessary but all the files are in the question 3 folder to run the code

Question 4:

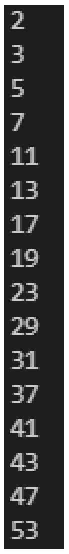
For question 4, we were asked for find prime numbers using two nested for loop.

The image below shows how I compiled the program.



```
jnguyen387@JimDesktop:/mnt/d/Libraries/Desktop/UCM_CSE_165_Lab_Submission/Lab 1/Jim_Nguyen/question4$ g++ primeNumber.cpp -o primeNumber
jnguyen387@JimDesktop:/mnt/d/Libraries/Desktop/UCM_CSE_165_Lab_Submission/Lab 1/Jim_Nguyen/question4$ ./primeNumber
```

After that, it will promptly print out all the prime numbers from 2-1000. I did not how high to put the range as it was not specified in the instructions, so I just made it 1000. Below is an image of the snippet of the output.



```
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
53
```

Here are just a written-out versions of the images above.

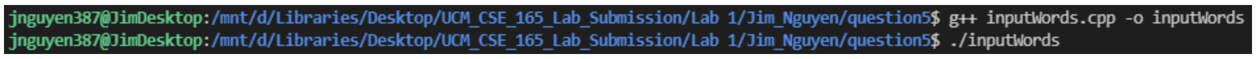
Compile: g++ primeNumber.cpp -o primeNumber

Run: ./primeNumber

Input: no input necessary

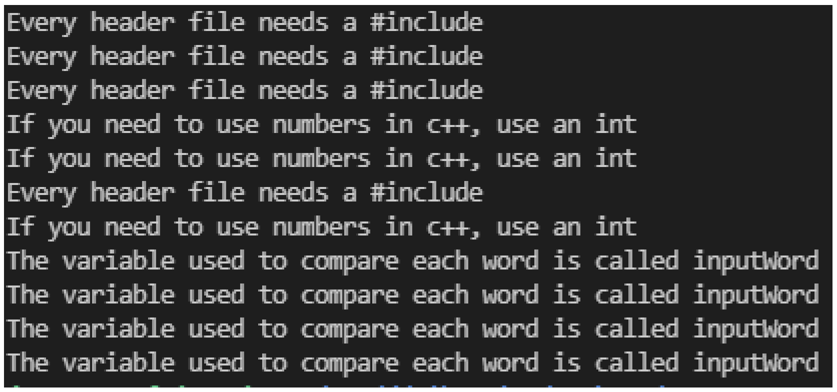
Question 5:

For question 5, we had to use an infinite while loop and read words from an input file. Then we decide what “interesting” words we wanted to use and use if else statements to assign those words numbers for the switch cases. Then we break once we decide the word to break the code. The image below shows how I compiled the program.



```
jnguyen387@JimDesktop:/mnt/d/Libraries/Desktop/UCM_CSE_165_Lab_Submission/Lab 1/Jim_Nguyen/question5$ g++ inputWords.cpp -o inputWords
jnguyen387@JimDesktop:/mnt/d/Libraries/Desktop/UCM_CSE_165_Lab_Submission/Lab 1/Jim_Nguyen/question5$ ./inputWords
```

After that, it promptly outputs the sentences that I have associated with the interesting words and quits the program once it has read the break word.



```
Every header file needs a #include
Every header file needs a #include
Every header file needs a #include
If you need to use numbers in c++, use an int
If you need to use numbers in c++, use an int
Every header file needs a #include
If you need to use numbers in c++, use an int
The variable used to compare each word is called inputWord
The variable used to compare each word is called inputWord
The variable used to compare each word is called inputWord
The variable used to compare each word is called inputWord
```

Here are just a written-out versions of the images above.

Compile: g++ inputWords.cpp -o inputWords

Run: ./inputWords

Input(file):inputWords.cpp (in question 5 folder)