CS 202 Homework 000

Kelby Hubbard

January 17, 2020

- Repository Link: `https://github.com/krhubbard2/CS202`
- Git Commits: `https://github.com/krhubbard2/CS202/commits`
- This homework took approximately 2 hours to complete.

1 Design

The designs on these were pretty straight forward. For the main program first I just made sure a generic command line argument worked and printed anything to the console. From their I integrated grabbing the strings, getting a double from them, and printing out the conversion.

2 Post Mortem

This assignment went very well. I had no problems getting the command line arguments to do what I want and I found argv[1] (or whichever) to work very well and exactly how I wanted it to. I would say the part that took me the longest was the error checking. I wanted to ensure it gave an error if you entered anything that wasn't a number, while still allowing a negative symbol in the front, and to ensure you didn't type something below absolute zero. After some checks, I believe this program is pretty polished and free of errors.

3 Answers to Questions

1. There are plenty of uses of software in science, medicine, and entertainment. One that pops into my mind first is that in medicine software allows us to keep track of many doctor notes and patient files very easily and cleanly. By having software that groups all notes to an individual patient it is very easy for any new doctor or hospital to look into the software and see exactly who they are working with.
2. I believe a software developer is anyone who thinks they can design a program that can make any part of our lives easier or more automated.
3. My computer has 16GB of main memory (RAM) and has 2 hard drives. It has a 1TB HDD and a 250GB SSD.
4. Five applications for computer programs would be personal, business, school, military, and robotics. The one that I find most interesting would be for business. There are many ways applications can be integrated in a business in which it can benefit a company. One day I would like to work for a big corporations designing software which would make the work load of the company easier and more manageable.
5. There are four binary digits in each hexadecimal digit, this is the same for decimal digits.

4 Program 1 Temperature

4.1 Temperature Sample Output

Listing 1: Sample Program Output

```
USER INPUT: ./temperature --ftoc 16.41
CONSOLE OUTPUT: -8.66111 celsius
USER INPUT: ./temperature --ctof -8.66111
CONSOLE OUTPUT:16.41 fahrenheit
```

```
USER INPUT: ./temperature --ftoc -425
CONSOLE OUTPUT:-253.889 celsius
USER INPUT: ./temperature --ftoc abc
CONSOLE OUTPUT: Improper entry. Please enter a valid
   temperature.
```

4.2 Git Commit Messages

Date	Message
2020-01-16	Created basis of program. Prints "ftoc" when command line is ran.
2020-01-16	Implemented c_ctof{const char* str} in main.
2020-01-16	Implemented cpp_ftoc{str} in main.
2020-01-16	Added error checking for absolute zero.
2020-01-16	Added bool to ensure entry is digits only. Gives error otherwise.
2020-01-16	Fixed error when user inserted negative after 1st character.
2020-01-16	Fixed an error where 8 would throw an error
2020-01-16	Updated main.o and temperature

4.3 Temperature Main

```
// Kelby Hubbard
// CS202
// Jan. 16, 2020
// Command Line Arguments

#include <iostream>
using std::cin;
using std::cout;
using std::endl;
#include <string>
using std::string;
#include <sstream>
using std::istringstream;

//Conversion using C++ STL
double cpp_ftoc(const char* str)
{
  istringstream iss(str);
  double fahrenheit;
  iss >> fahrenheit;;

  double celsius = (fahrenheit - 32)*5/9;

```

```
  return celsius;
}
//Conversion using C Function strtod
double c_ctof(const char* str)
{
  double celsius = strtod((str), NULL);
  double fahrenheit = (9*celsius/5)+ 32;

  return fahrenheit;
}
//Ensures string entered are digits only
bool digits(const string s)
{
  return s.find_first_not_of("0123456789.") == string::npos;
}
int main(int argc, const char** argv)
{
  //Fahrenheit to celius
  if (argc >= 2 && argv[1] == string("--ftoc"))
  {
    //Checks if temp entered is below absolute zero
    if (cpp_ftoc(argv[2]) < -273.15)
    {
      cout << "Error. That temperature is below absolute zero.\n";
    }

    //Checks if negative number
    else if (argv[2][0] == '-')
    {
      //Erases negative sign
      string neg = argv[2];
      neg.erase (neg.begin() + 0);
      //Ensures number is only digits
      if (digits(neg))
      {
        cout << cpp_ftoc(argv[2]) << " celsius" << endl;
      }
      //Improper entry
      else
      {
        cout << "Improper entry. Please enter a valid temperature." << endl;
      }
    }
    //Entry only contains digits
    else if (digits(argv[2]))
    {
    cout << cpp_ftoc(argv[2]) << " celsius" << endl;
    }
    //Improper entry
    else
    {
      cout << "Improper entry. Please enter a valid temperature." << endl;
    }
  }

  //Celcius to fahrenheit
  else if (argc >= 2 && argv[1] == string("--ctof"))
  {
    //Checks if temp entered is below absolute zero
```

```
87      if (c_ctof(argv[2]) < -459.67)
88      {
89        cout << "Error. That temperature is below absolute zero.\n";
90      }
91      //Checks if negative
92      else if (argv[2][0] == '-')
93      {
94        //Delete negative
95        string neg = argv[2];
96        neg.erase (neg.begin() + 0);
97        //Checks now that negative is gone that there is only digits
98        if (digits(neg))
99        {
100         cout << c_ctof(argv[2]) << " fahrenheit" << endl;
101       }
102       //Impropery Entry
103       else
104       {
105         cout << "Improper entry. Please enter a valid temperature." << endl;
106       }
107     }
108     //Only digits entered
109     else if (digits(argv[2]))
110     {
111     cout << c_ctof(argv[2]) << " fahrenheit" << endl;
112     }
113     //Error for improper entry.
114     else
115     {
116       cout << "Improper entry. Please enter a valid temperature." << endl;
117     }
118   }
119
120
121
122   return 0;
123 }
```

---

5 Gutenberg Excerpt

5.1 Gutenberg Excerpt Sample Output

Listing 2: Sample Program Output

```
'Heathcliff--I shudder to name him! has been a stranger in
   the house from last Sunday till to-day.  Whether the
   angels have fed him, or his kin beneath, I cannot tell;
   but he has not eaten a meal with us for nearly a week.
   He has just come home at dawn, and gone up-stairs to his
    chamber; locking himself in--as if anybody dreamt of
   coveting his company!  There he has continued, praying
```

```
like a Methodist: only the deity he implored is
senseless dust and ashes; and God, when addressed, was
curiously confounded with his own black father!  After
concluding these precious orisons--and they lasted
generally till he grew hoarse and his voice was
strangled in his throat--he would be off again; always
straight down to the Grange!  I wonder Edgar did not
send for a constable, and give him into custody!  For me
, grieved as I was about Catherine, it was impossible to
 avoid regarding this season of deliverance from
degrading oppression as a holiday.
```

5.2 Git Commit Messages

Date	Message
2019-01-16	Succesfully reads file and prints excerpt.

5.3 Gutenberg Main

```
1  // Kelby Hubbard
2  // CS202
3  // January 16, 2020
4  // Gutenberg Excerpt
5
6  #include <iostream>
7  using std::cout;
8  using std::endl;
9  #include <string>
10 using std::string;
11 using std::getline;
12 #include <fstream>
13 using std::ifstream;
14 #include <sstream>
15 using std::istringstream;
16
17 int main()
18 {
19   ifstream fin("ebookplaintext.txt");
20
21   //Can it read file?
22   if (!fin)
23   {
24     cout << "Can't open file." << endl;
25   }
26   else
27   {
28     while(true)
29     {
30       string line;
```

```
      getline(fin, line);
      //eof checking
      if (!fin)
      {
        if (fin.eof())
        {
          return true;
        }
        else
        {
          return false;
        }
      }
      //Grab each line word by word
      istringstream iss(line);
      string word;
      iss >> word;

      //Start of the excerpt
      if (word == "'Heathcliff--I")
      {
        //Print out excerpt until blank line
        do {
          cout << line << " ";
          getline(fin, line);
        } while(line != "");
        cout << endl;
      }
    }
  }

  return 0;
}
```