**Unit Project Two:**

Unit Conversions

Units used

Length: meter, centimeter, foot, inch

Mass: kilogram, gram, slug

Time: hour, minute, second

Angle: degree, radian, grade

**PROGRAM:**

All the functions are in **Bold** and all the comments are Highlighted for legibility.

**MYLIBRARY.H:**

The library header is 26 lines long

//Function declarations

//Function to get unit data into one string

string &**consolidate** (string &z1, string &z2, string &z3);

//Function to change all chars in a string to lowercase

string &**lowercase** (string &s);

//Function to change the first char in a string to uppercase

string &**Capitalize** (string &s1);

//Function that takes each char in a string and puts it in a char

void **String2Char** (string, char&, char&, char&);

//Function to get data for the loop

void **GetData** (double&, double&, double&, double&, double&);

// Function to return the conversion factor

double **convFactor** (string);

//Function to print the table

void **printTable** (double, double, double, double, double, double, string, string, string);

//Function to clear the screen and go back to the menu

void **goHome** ();

**MAIN:**

The main is 155 lines long

// \_\_\_\_ \_\_ \_\_

// / / \ / \

// / / /

// / \\_\_\_/ \\_\_\_/

//

// Date: 10/26/2021

// Name: David Vermaak

//Project Description: The main function of Unit Project 2: Unit Conversion

// Inputs: units, initial and final values, formatting values, etc.

// Outputs: a formatted table

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <iostream> //This header containing cout and cin

#include <math.h> //This header allows for the use of more complex mathematical operators

#include <string> //This header enables string functions

#include <cctype> //This header contains functions that give Boolean answers for given types of char

#include <iomanip> //This header contains functions to better format output

#include <ctype.h> //This header contains functions to manipulate chars

#include <conio.h> //This header declares getch which pauses until a key is pushed

#include "MyLibrary.h" //Personal library header

using namespace std; //introduces namespace std

int **main** () //The Main function

{

string s1, s2, s3, C; //initializing strings

double f, a, a0, a1, a2, a3, a4; //initializing doubles

do //do while loop to make the program repeatable

{

// display menu

cout << " Hello, this program outputs a formatted list of unit conversions\n"

<< " Please enter one of the following options to continue:\n"

<< " Length \n" << " Mass \n" << " Time \n" << " Angle \n" << " Quit\n\n";

//get user input

cin >> s1;

**lowercase**(s1); //calls the function lowercase to edit a string s1

if (s1 == "quit") return 0; //quit option of the menu

else if (s1 == "length") //length option of the menu

{

cout << " You have chosen Length conversion\n"

<< " Please choose your initial unit:\n"

<< " Meters: (m) Centimeters: (cm) Feet: (ft) or Inches: (in)\n";

cin >> s2;

cout << " Please choose your second unit:\n"

<< " Meters: (m) Centimeters: (cm) Feet: (ft) or Inches: (in)\n";

cin >> s3;

//All of the following functions are declared in the MyLibrary.h file and defined in the MyLibrary.cpp file

// Both of the Files are in the UnitProject2 folder

// These functions are reused at least four times throughout the program

C = **consolidate** (s1, s2, s3); //calls the function consolidate and stores it to a string

**lowercase**(C); //calls the function lowercase to edit a string C

f = **convFactor**(C); //calls the function convFactor and stores it to the double f

if (f>0) //Fail condition if -1.0 is returned from **convFactor**

{

**GetData**(a0, a1, a2, a3, a4); //calls the function GetData

**printTable**(a0, a1, a2, a3, a4, f, s1, s2, s3); //calls the function printTable

}

**goHome**(); //calls the function goHome

}

else if (s1 == "mass") //Mass option of the menu

{

cout << " You have chosen Mass conversion\n"

<< " Please choose your initial unit:\n"

<< " Kilogram: (kg) Gram: (g) or Slug: (sl)\n";

cin >> s2;

cout << " Please choose your second unit:\n"

<< " Kilogram: (kg) Gram: (g) or Slug: (sl)\n";

cin >> s3;

C = **consolidate** (s1, s2, s3);

**lowercase**(C);

f = **convFactor**(C);

if (f>0)

{

**GetData**(a0, a1, a2, a3, a4);

**printTable**(a0, a1, a2, a3, a4, f, s1, s2, s3);

}

**goHome**();

}

else if (s1 == "time") //Time option of the menu

{

cout << " You have chosen Time conversion\n"

<<" Please choose your initial unit:\n"

<<" Hour: (hr) Minute: (min) or Second: (sec)\n";

cin >> s2;

cout << " Please choose your second unit:\n"

<< " Hour: (hr) Minute: (min) or Second: (sec)\n";

cin >> s3;

C = **consolidate** (s1, s2, s3);

**lowercase**(C);

f = **convFactor**(C);

if (f>0)

{

**GetData**(a0, a1, a2, a3, a4);

**printTable**(a0, a1, a2, a3, a4, f, s1, s2, s3);

}

**goHome**();

}

else if (s1 == "angle") //Angle option of the menu

{

cout << " You have chosen Angle conversion\n"

<< " Please choose your initial unit:\n"

<< " Degree: (deg) Radian: (rad) or Grade: (g)\n";

cin >> s2;

cout << " Please choose your second unit:\n"

<< " Degree: (deg) Radian: (rad) or Grade: (g)\n";

cin >> s3;

C = **consolidate** (s1, s2, s3);

**lowercase**(C);

f = **convFactor**(C);

if (f>0)

{

**GetData**(a0, a1, a2, a3, a4);

**printTable**(a0, a1, a2, a3, a4, f, s1, s2, s3);

}

**goHome**();

}

else //The failure condition and error message for an incorrect input

{

cout << " Incorrect Entry\n\n ";

**goHome**();

}

} while (s1 != "quit" ); //other part of the while loop

return 0;

}

**MYLIBRARY.CPP:**

The library is 242 lines long

#include <iostream> //This header containing cout and cin

#include <conio.h> //This header declares getch which pauses until a key is pushed

#include <math.h> //This header allows for the use of more complex mathematical operators

#include <string> //This header enables string functions

#include <cctype> //This header contains functions that give Boolean answers for given types of char

#include <iomanip> //This header contains functions to better format output

#include <ctype.h> //This header contains functions to manipulate chars

#include "MyLibrary.h" //Personal library header

//Functions

//Function gets data from the user and then stores it in doubles

void **GetData** (double& A0, double& A1, double& A2, double& A3, double& A4)

{

cout << " Enter the starting value (eg: 0)\n";

cin >> A0;

cout << " Enter the ending value (eg: 10)\n";

cin >> A1;

cout << " Enter the increment (eg: 1)\n";

cin >> A2;

cout << " Enter the number of digits for the 1st unit (eg: 1)\n";

cin >> A3;

cout << " Enter the number of digits for the 2nd unit (eg: 3)\n";

cin >> A4;

if (A1>=A0 && A1>=A2 && A3>=0 && A4>=0) return; //checks for mathematically impossible entries

else cout << " Error: value outside of scope"; //error message

}

//Function to change all chars in a string to lowercase

string &**lowercase** (string &S1)

{

for (int i = 0; i<S1.length(); i++)

{ S1[i] = tolower(S1[i]);} //uses tolower in a for loop

return S1;

}

//Function to change the first char in a string to uppercase

string &**Capitalize** (string &s1)

{ s1[0] = toupper(s1[0]); //uses toupper for capitalization

return s1; }

//Function to get data into one string

string &**consolidate** (string &S1, string &S2, string &S3)

{

string C = S1.substr (0,1)+ S2.substr (0,1)+ S3.substr (0,1); //uses .substr to add strings together

return C;

}

// Function to return the conversion factor as a double, from the data in a string

double **convFactor** (string z)

{

char c0, c1, c2; //initializes the chars

**String2Char** (z, c0, c1, c2); //uses function **String2Char**

//this is a massive if/else if function to return the correct numeric value from the 30 options

//each option is identified by a three-char designation for example: LMF = LengthMeterFoot

//12 for Length: lmc lmf lmi lcm lcf lci lfc lfm lfi lic lim lif

if (c0 == 'l')

{

if (c1 == 'm') //meter

{

if (c2 == 'c'){return 0.01;} //char code lmc

else if (c2 == 'f'){return 0.3048;} //char code lmf

else if (c2 == 'i'){return 0.0254;} //char code lmi

}

else if (c1 == 'c') //centimeter

{

if (c2 == 'm'){return 100.0;}

else if (c2 == 'f'){return 30.48;}

else if (c2 == 'i'){return 2.54;}

}

else if (c1 == 'f') //foot

{

if (c2 == 'm'){return 3.28084;}

else if (c2 == 'c'){return 0.0328084;}

else if (c2 == 'i'){return 12.0;}

}

else if (c1 == 'i') //inch

{

if (c2 == 'm'){return 39.37008;}

else if (c2 == 'f'){return 12.0;}

else if (c2 == 'c'){return 0.3937008;}

}

}

//6 for Mass: mkg mks mgk mgs msk msg

else if (c0 == 'm')

{

if (c1 == 'k') //kilogram

{

if (c2 == 'g'){return 0.001;}

else if (c2 == 's'){return 14.5939;}

}

else if (c1 == 'g') //gram

{

if (c2 == 's'){return 14593.9;}

else if (c2 == 'k'){return 1000.0;}

}

else if (c1 == 's') //slug

{

if (c2 == 'k'){return 0.06852177;}

else if (c2 == 'g'){return 6.852177e-5;}

}

}

//6 for time: thm ths tms tmh tsm tsh

else if (c0 == 't')

{

if (c1 == 'h') //hour

{

if (c2 == 'm'){return (1.0/60.0);}

else if (c2 == 's'){return (1.0/3600.0);}

}

else if (c1 == 'm') //minute

{

if (c2 == 's'){return (1.0/60.0);}

else if (c2 == 'h'){return 60.0;}

}

else if (c1 == 's') //second

{

if (c2 == 'h'){return 3600.0;}

else if (c2 == 'm'){return 60.0;}

}

}

//6 for angle: adr adg arg ard agr agd

else if (c0 == 'a')

{

if (c1 == 'd') //degree

{

if (c2 == 'r'){return 57.29578;}

else if (c2 == 'g'){return 0.9;}

}

else if (c1 == 'r') //radian

{

if (c2 == 'g'){return 63.661977;}

else if (c2 == 'd'){return 0.01745329;}

}

else if (c1 == 'g') //grade

{

if (c2 == 'd'){return (1.0/0.9);}

else if (c2 == 'r'){return 0.015708;}

}

}

else

{ cout << "Error, Units not recognized"; //error message

return -1.0; //returns a -1.0 which causes an if statement to fail

}

}

//Function that takes each char in a string and puts it in a char

void **String2Char** (string y, char& C0, char& C1, char& C2)

{

C0 = y[0];

C1 = y[1];

C2 = y[2];

}

//Function to print the table with data from **Main**, **convFactor,** and **GetData** as doubles and strings

void **printTable** (double Ar0, double Ar1, double Ar2, double Ar3, double Ar4, double q, string q1, string q2, string q3)

{ **Capitalize**(q1); **Capitalize**(q2); **Capitalize**(q3); //uses function **Capitalize**

cout << "\n\n\n\n The " << q1 << " Conversion of:\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\n"

<< right <<setw(8) << q2 << " to " << right << setw(8) << q3 <<endl;

for (float i = Ar0; i <= Ar1; i = i + Ar2)

{

cout <<"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n"

<< fixed << setprecision(Ar3) << right << setw(8) << i << " | "

<< setprecision(Ar4) << right << setw(8) << i/q << " |" <<endl;

}

}

//Function to clear the screen and go back to the menu

void **goHome** ()

{

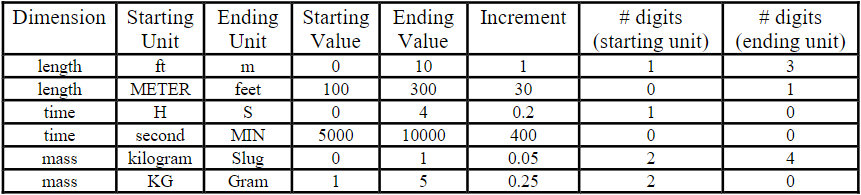
cout << "\n\n To return to the menu press enter";

getch(); //pauses until a key is pressed

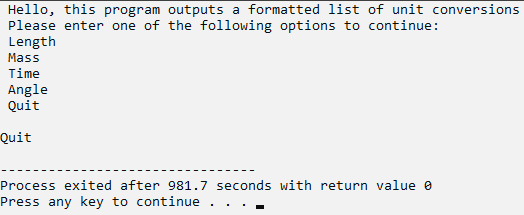
system("cls");

}

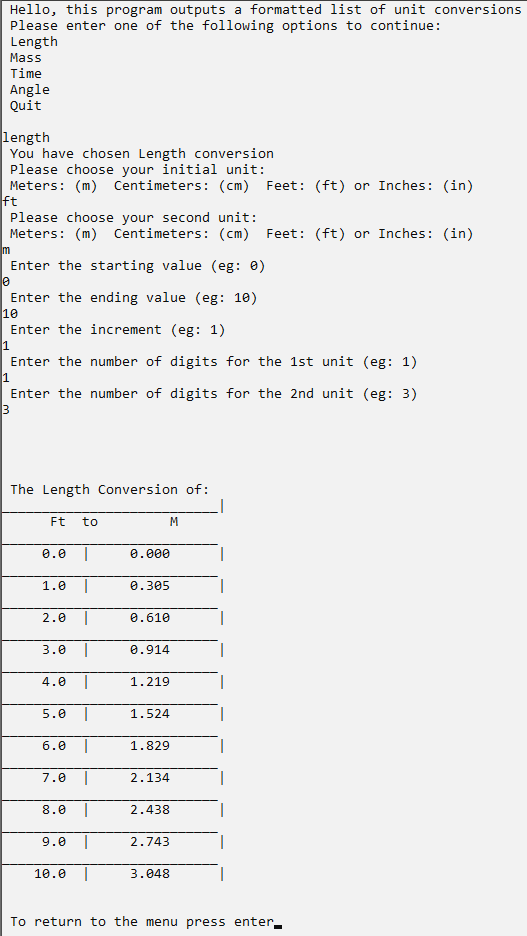
* Test Results: Run the program for the following test cases:



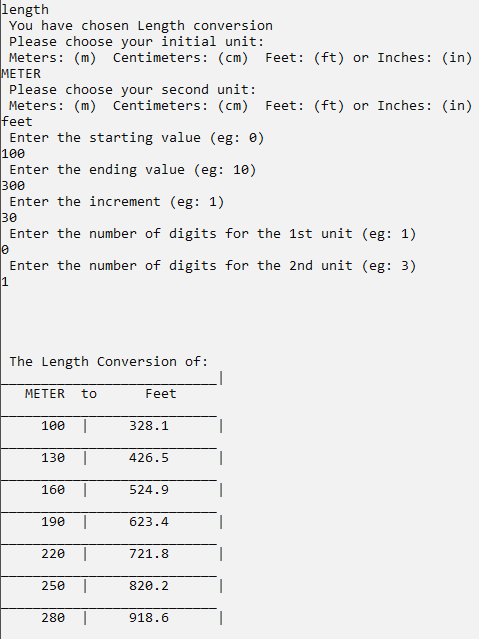
All 8 test cases were run together using the looping menu system. It was exited with this on the display:



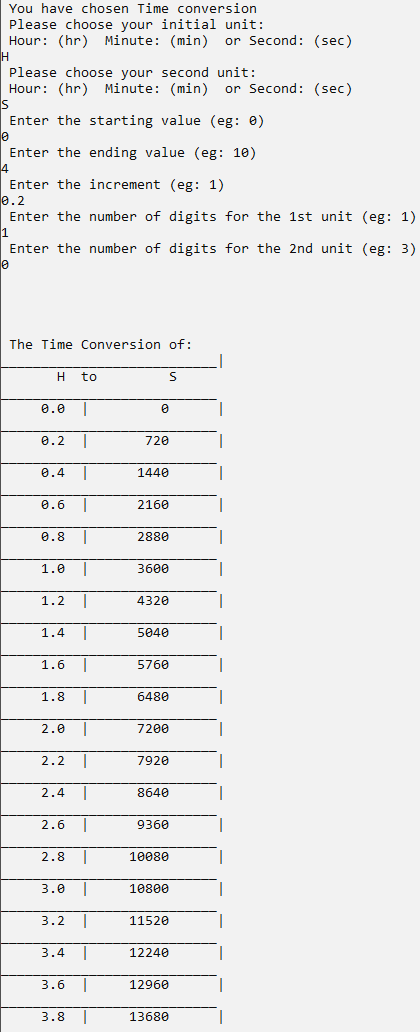
**Case 1:**



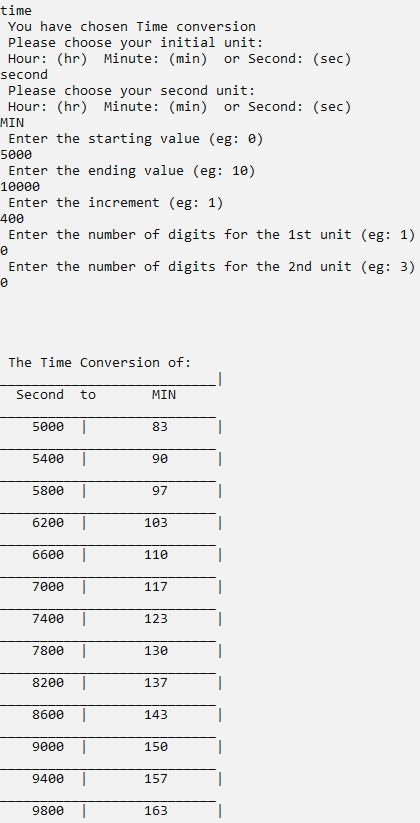
**Case 2:**



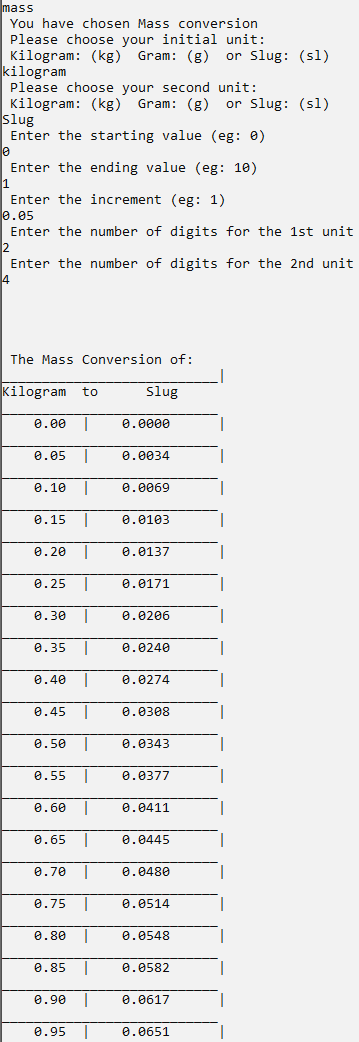
**Case 3:**



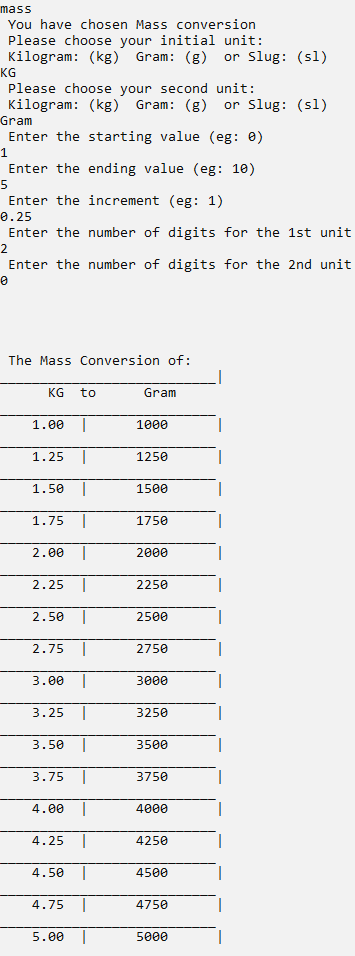
**Case 4:**



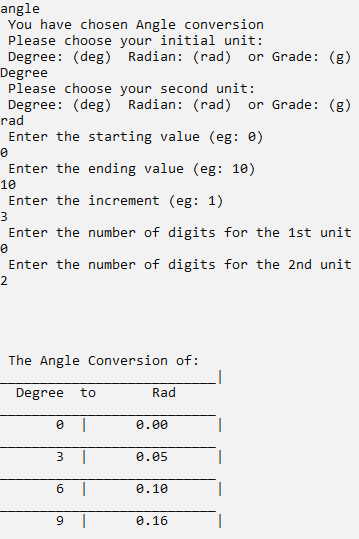
**Case 5:**



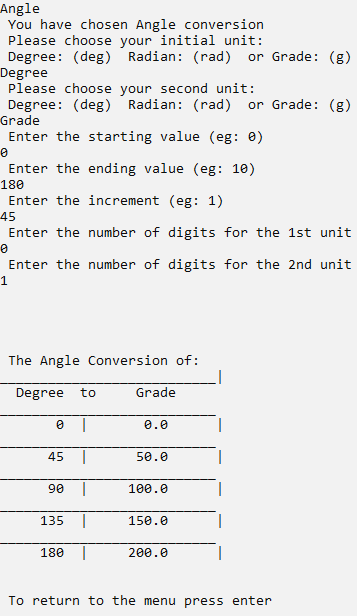
**Case 6:**



**Case 7:**



**Case 8:**



Program performance:

*The program’s performance was adequate. It compiled in 0.75 seconds and it is possible to run through and quit in under 30 seconds, with no noticeable lag. The UI, while barebones, conveys the information to the user quickly and adequately*.

Extra credit features include:

1. *Added additional dimensions and units. (Angle [degree, radians, grade])*

*2. Added ‘grid’ lines to divide the rows and columns of the table.*

*3. Used additional well-designed functions:*

**consolidate** //Function to get unit data into one string

*(Great function as it is used often, which shortens the program)*

**lowercase** //Function to change all chars in a string to lowercase ***(was required)***

*(Great function as it is used often and removes the need to add conditions to true/false statements due to case sensitivity. I definitely will be reusing this in all of my projects)*

**Capitalize** //Function to change the first char in a string to uppercase

*(I definitely will be reusing this in all of my projects)*

**String2Char** //Function that takes each char in a string and puts it in a char

*(Useful if data is stored in a string as only chars can be used for testing equalities e.g., if statements)*

**GetData** //Function to get data for the loop

*(Great function as it is used often, which shortens the program)*

**convFactor** // Function to return the conversion factor ***(was required)***

*(Possibly the largest function I will ever write, there has got to be a better way to do it)*

**printTable** //Function to print the table

*(Great function as it is used often, which shortens the program)*

**goHome** //Function to clear the screen and go back to the menu

*(Great function as it is used often, which shortens the program, I definitely will be reusing this in all of my projects)*

4*. Implemented the assignment as a DevC++ ‘project’ with separate files for the main, functions, and header.*

Potential improvements:

*There are 4 else if loops in the Main that contain most of the same information and functions, I think it could be possible to reduce the Main down to just a do while loop, an introduction, and a trio of if, else if, and else statements.*

*Pseudo-code:*

*do{ “introduction”*

*if (quit) return 0;*

*else if {****body*** *function}*

*else{“error message”;* ***goHome****();}*

*}while(;*

*// body function*

*Void* ***body*** *(string* S1*)*

*{ “you have chosen”* S1 *“conversion\n Please enter the initial unit :\n”*

*C =* ***consolidate*** *(s1, s2, s3);*

***lowercase****(C);*

*f =* ***convFactor****(C);*

*if (f>0)*

*{****GetData****(a0, a1, a2, a3, a4);*

***printTable****(a0, a1, a2, a3, a4, f, s1, s2, s3);}*

***goHome****();*

*}*

*The only problem with this is recursion, as it would be three functions deep at some points and I don’t know if it would work. Additionally, I doubt it would reduce the total length of the program as the lines saved in Main would just be moved to the already bloated* ***MyLibrary*** *file. The thing I am most interested in is how to reduce the size of the* ***convFactor*** *function which is at a whopping 140 lines, 58% of the whole file, while being only one of the eight functions.*