Victoria Croffie
CS 604
Project 1-Report
09/27/12

Problem Specification

The problem specification is to complete a program that converts input temperatures from Fahrenheit into Celsius. On top of that any input less than Absolute Zero must have an error output. The program has already been designed and partially coded so all that is needed is to finish the code and make sure it compiles without errors and is up to quality specification.

~TEST PLAN~
Objective – in where a plan gets tested

TEST OBJECTIVE	FAHRENHEIT INPUT	CELSIUS OUTPUT
Easy-to-calculate input	32.000000	0.000000
Another legal input	212.000000	100.000000
Minimum legal input	-459.670000	273.594444
Out-of-range input	-459.680000	INPUT ERROR!

Synopsis

The above is the test plan for a Fahrenheit to Celsius program. The first test objective involves an easy to calculate input which will be 32.000000 which converts to 0.000000 as the output. Since the variables involved in the program are doubles, there are 6 digits after the decimals. Next we have another legal input which is 212.000000, the output for this is 100.000000. The minimum legal input for this program is -459.670000 which the output for that is 273.594444. And finally the out-of-range input is -459.680000 which computes to an input error output. When these inputs are put through the program you get the same outputs and so the that means that the test plan went according to plan.

Victoria Croffie
CS 604
Project 1-Report
09/27/12

Manual

The basis of my program is a simple Fahrenheit to Celsius temperature conversion. After compiling it, all that needs to be done is run the file which will then have you input whatever temperature you desire to know the conversion of. While the comments in my code should help clear up any confusion, this step by step should also help complement it. It is to be noted that I have adopted the Compact Control Readability style as this style is most akin to the English language format as well as making it easy to point out control statements and coding blocks.

The first line of the code is the standard #include <stdio.h>. This is simply the library that you need for a c code to function. The int main(void) is a calling function which indicates what type of return that the code will have. The (void) aspect simply declares that the machine is suppose to ignore command line arguments. Moving on we now have our declarations: double Absolute_Zero = -459.67, double fahr = 0.0, and double cels which will be used for our variables.

Now comes the meat of the code. We have two puts statements which is simply a way to set up what this code will be about and add nice spacing as well. The printf functions that comes after is the first statement that give the user a direction on how they should enter the input. The scanf simply scans and stores the input in the computer's memory to be used later on. “%lg” is used because it is a double variable as opposed to any other type. This also dictates home much space is needed to store the input.

We now come across a while loop which is in place to disallow any input lower than a certain number to be used. If it is used, the user will get an error message as stated in the following printf statement and will then be made to put in another number for computation. This new number will now be scanned and stored using the scanf function.

Now if you have already entered an acceptable number or you have re-entered one after getting the error message, your input will be stored in the memory. We will then have a printf function that states what number you have entered. We now come to the math of the code and the first use of the Celsius variable. We enter the equation for converting a temperature in Fahrenheit to Celsius, making sure to use rational numbers as opposed to whole numbers as whole number will be viewed as a string and not computed properly.

The computer now inputs your input into the equation to calculate the answer. You will then use the printf function to display the resulting answer with printf("\n Temperature in Celsius is: %lf\n", cels). After another printf function thanking the user for using this program, the program will then end. The return(0) at the end simply means that the program is a success.

This basically is the gist of the program and what type of statements are needed for the program. For all other details be sure to look at the code itself and the comments that accompany it.