

## Instructions

For this programming assignment, you will create a small program that contains four functions, including the main function. This program will provide unit conversions from degrees Fahrenheit to Celsius, Celsius to Fahrenheit, and pounds to kilograms.

On startup, the user should be prompted with a ‘menu’ of options from which to select the desired conversion. Each menu option will represent one of the functions you are to implement, with an additional option to exit the program.

If the user enters anything not tied to an option, they should be notified that the option entered is invalid and prompted to try again. A user should be able to enter as many options as they would like, with the program only exiting once they enter the ‘exit’ option.

## Grading

**ALL** of your code should be placed inside of the generated **main.cpp** file created by your CLion IDE when you create a new project. Only libraries native to C++ are allowed to be used.

**Note:** C++ 20 will be used to grade this and all C++ programming assignments, so be sure your code works on this version.

Code that does not compile will incur an automatic 50% penalty. Similarly, a program entering an infinite loop, encountering a runtime error, or freezing will incur an automatic 25% penalty. Each function is worth 20 points (80 points total). The final 20 points will come from relevant documentation throughout your program.

Documentation will be graded as follows. Every function you create should be as fully documented as possible. Full documentation is placed directly above a function’s implementation and includes, but is not limited to: 1) listing any and all parameters describing what they mean and their range of acceptable/expected values, 2) describe the return type and what the value means, and 3) describe the expected behavior of the function.

A basic documentation example is provided above the sample code output at the bottom of this document.

As a general rule: *If you are unsure whether a comment should be there or not, put one.*

Once you have completed writing your solution code, take a screenshot of your program running **before the user has selected an option**. ZIP the **screenshot and your main.cpp** file into a folder named ‘PA1\_ULID’ replacing ‘ULID’ with your own ULID. Upload the ZIP folder to Moodle for submission.

### Function 1: main

This is the menu function of your program and should operate in a loop until the user opts to exit. It should prompt the user to select a unit conversion that they wish to have calculated, with a fourth option being to exit. You may determine the type of input used for this selection – a good choice would be to have the user enter a number ranging from 1 to 4, with 4 being the exit option.

If the user enters an invalid option, print that their selection was invalid and begin the loop again, prompting them for a new selection.

Once a valid conversion selection has been made, prompt the user for the numeric value to be converted – and store this in a variable of type double. Next, call the appropriate function based on their selection, which should return the converted numeric value.

Use a **switch statement** to process the user selection, get numeric input, call the desired conversion function, and print the conversion results. Name the other three functions as appropriate.

**Function 2: Fahrenheit to Celsius**

This function will convert degrees Fahrenheit to degrees Celsius using the following formula:

$$C = (F - 32) * (5/9.0)$$

Once the input value has been converted, return the value to your main function and print the conversion there.

**Function 3: Celsius to Fahrenheit**

This function will convert degrees Celsius to degrees Fahrenheit using the following formula:

$$F = C * (9/5.0) + 32$$

Once the input value has been converted, return the value to your main function and print the conversion there.

**Function 4:**

This function will convert a weight in pounds to its weight in kilograms. One pound is equal to 0.45359237 kilograms.

Once the input value has been converted, return the value to your main function and print the conversion there.

### Documentation Example

```
/* This Function Computes x!
 * Precondition:
 *     parameter x is int the value you want the factorial of. x >= 0 && x! <= INT_MAX
 * Postcondition:
 *     Function value == x!
 */
int Factorial(int x) {
    int result = 1; //holds partial product
    while(x > 0) {
        result = result * x--;
    }
    return result;
}
```

### Sample Code Output

To begin, select from the following options:

Enter '1' to convert from degrees Fahrenheit to Celsius.

Enter '2' to convert from degrees Celsius to Fahrenheit.

Enter '3' to convert weight from pounds to kilograms.

Enter '4' to exit.

Selection: 3

To convert from pounds to kilograms, enter a weight in pounds: 100

-> 100 pounds equals 45.359237 kilograms.

To begin, select from the following options:

Enter '1' to convert from degrees Fahrenheit to Celsius.

Enter '2' to convert from degrees Celsius to Fahrenheit.

Enter '3' to convert weight from pounds to kilograms.

Enter '4' to exit.

Selection: moodle

I'm sorry, "'moodle'" is not a valid menu selection. Please try again.

To begin, select from the following options:

Enter '1' to convert from degrees Fahrenheit to Celsius.

Enter '2' to convert from degrees Celsius to Fahrenheit.

Enter '3' to convert weight from pounds to kilograms.

Enter '4' to exit.

Selection: \_