Activity No. 6.2	
Built-in Functions	
Course Code: CPE007	**Program:** Computer Engineering
Course Title: Programming Logic and Design	**Date Performed:10-28-25**
Section:CPE11S1	**Date Submitted:10-30-25**
Name(s):JOEL A ALMIROL	**Instructor: Engr. Jimlord M. Quejado**
6. Output	

Activities:6.2

Note: ***Perform each of these steps for the below problems:***

- ***Read the problem statement***
- ***Write a C program***
- ***Test, Debug and execute the C program***
- ***Create a screenshot of the code and output and submit it using the activity template***

1. **Create a program that defines a function to compute for the volume of a cube. The formula of the volume of the cube is given as V = s * s * s.**
2. **Define a function hypotenuse that calculates the length of the hypotenuse of a right triangle when the other two sides are given. Use this function in a program to determine the length of the hypotenuse for each of the following triangles. The function takes two arguments of type double and return the hypotenuse as a double.**
3. **Implement the following integer functions:**
    1. **Function celsius returns the Celsius equivalent of a Fahrenheit temperature.**
    2. **Function fahrenheit returns the Fahrenheit equivalent of a Celsius temperature.**
    3. **Use these functions to write a program that prints charts showing the Fahrenheit equivalents of a Celsius temperatures from 0 to 100 degrees, and the Celsius equivalents of all Fahrenheit temperatures from 32 to 212 degrees. Print the outputs in a neat tabular format that minimizes the number of lines of output while remaining readable.**

function 1.cpp

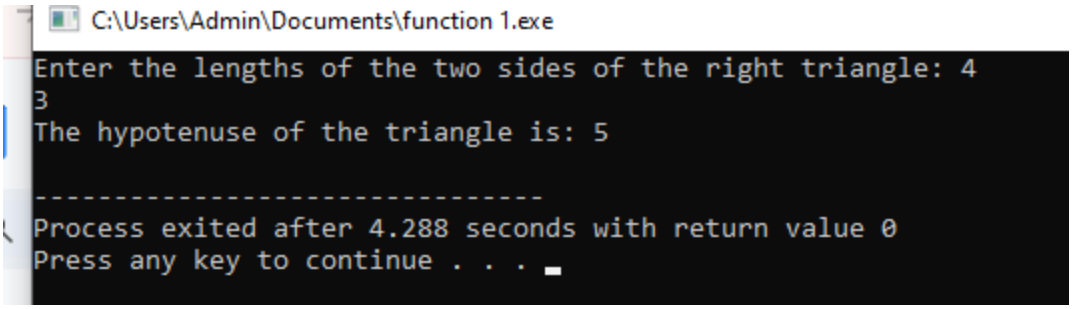
```cpp
#include <iostream>
using namespace std;

// Function prototype
double cubeVolume(double side);

int main() {
    double side, volume;

    cout << "Enter the side length of the cube: ";
    cin >> side;

    volume = cubeVolume(side);

    cout << "The volume of the cube with side " << side
         << " is: " << volume << endl;

    return 0;
}

// Function definition
double cubeVolume(double side) {
    return side * side * side;
}
```

C:\Users\Admin\Documents\function 1.exe

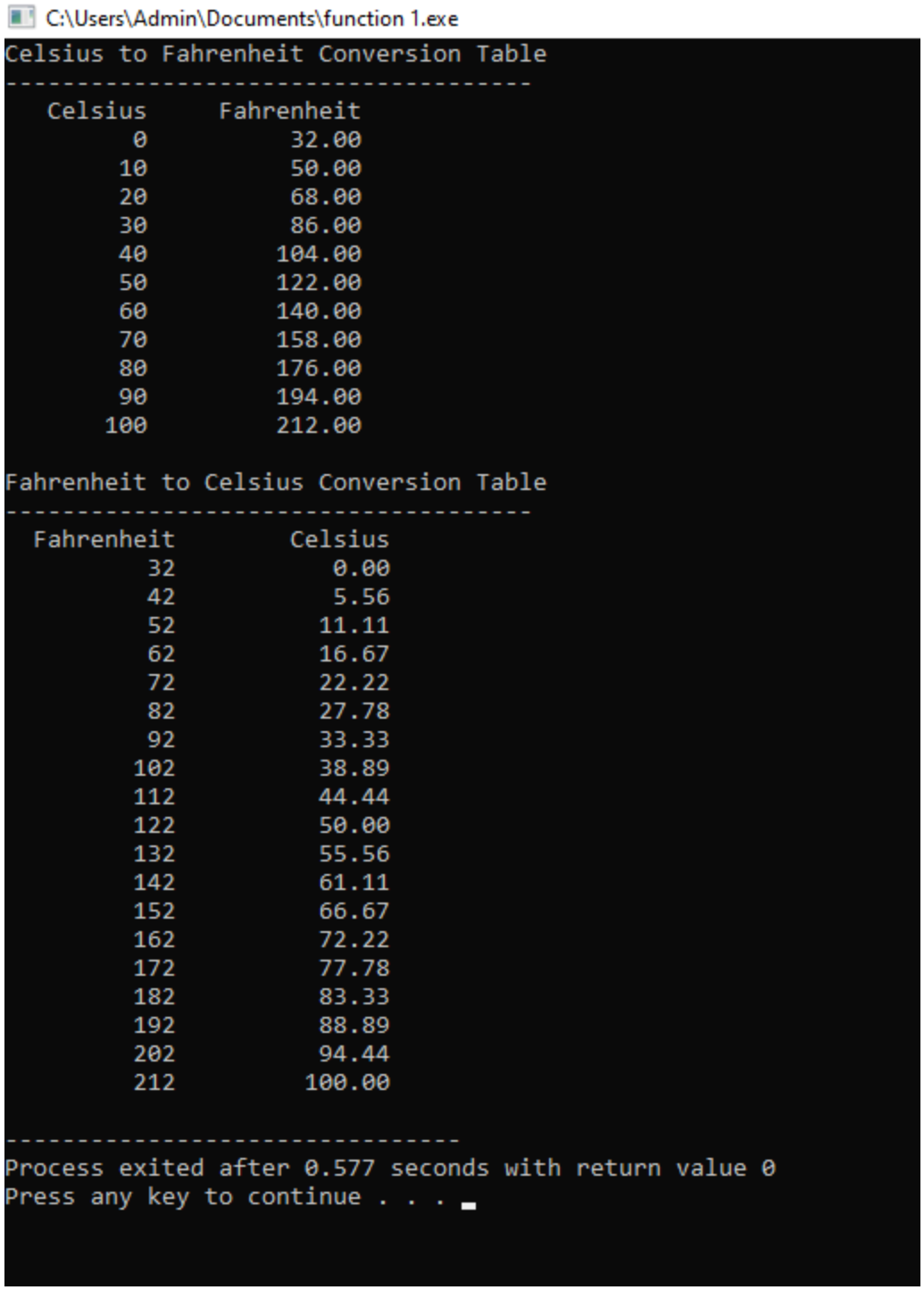
```
Enter the side length of the cube: 3
The volume of the cube with side 3 is: 27

--------------------------------
Process exited after 13.95 seconds with return value 0
Press any key to continue . . .
```

function 1.cpp

```cpp
#include <iostream>
#include <cmath> // for sqrt()
using namespace std;

// Function prototype
double hypotenuse(double side1, double side2);

int main() {
    double a, b, h;

    cout << "Enter the lengths of the two sides of the right triangle: ";
    cin >> a >> b;

    h = hypotenuse(a, b);

    cout << "The hypotenuse of the triangle is: " << h << endl;

    return 0;
}

// Function definition
double hypotenuse(double side1, double side2) {
    return sqrt(side1 * side1 + side2 * side2);
}
```




```
    return 0;
}

// Function definitions
double celsius(double fahrenheit) {
    return (5.0 / 9.0) * (fahrenheit - 32);
}

// NOTE: The definition for fahrenheit(double celsius) is declared but not fully shown in the image.
// Based on standard conversion, it would typically be:

double fahrenheit(double celsius) {
    return (9.0 / 5.0) * celsius + 32;
}
```



7. Supplementary Activity

Analysis: In this exercise, the goal is to calculate the volume of a cube using a function that is defined by the user. First, the program gets the user to enter the length of one side of the cube, then calls a function called cubeVolume() to actually compute the volume. The function body computes the volume of the cube using the simplified formula of V=s×s×s (V=s3). The program also outputs the value the user entered, and it outputs the result of the calculation. This exercise teaches how to write and call a function in C++, define function arguments, and return a value from a function. By separating the calculation into its own function, the overall program becomes more organized and reusable. The variable type double is used to provide more accurate results with

decimal input values. Overall, this simple program shows how to do simple arithmetic, read user input from the keyboard, write output to the screen, and use a program structure to keep the program modular, by using function prototypes — all part of programming.

Analysis: This application utilizes a custom function hypotenuse() to calculate the hypotenuse length of a right triangle. The user is asked to provide the lengths for the two shorter sides (commonly called the base and height). The function will take the lengths provided by the user and apply the Pythagorean theorem ℏ=??2+??2h=a2+b2 to find the hypotenuse length. In this program the library is used for the sqrt() function which calculates the square root of the sum of the squares.
This program serves to demonstrate how math formulas translate to C++ using functions and libraries in the standard namespace. The calculation logic has been segmenting into the hypotenuse() function, which will enhance readability and maintenance of the program. It also provides another opportunity to see function prototypes, effective data handling (data type double), and formatted output. This assignment is a great opportunity to put math to programming to a useable application.

Analysis: In this exercise, the program applies the two functions celsius() and fahrenheit() to convert temperature from Celsius to Fahrenheit and vice versa. The first function will convert Fahrenheit to Celsius, utilizing the formula C = (F - 32) x 5/9, while the second function will do the opposite, converting Celsius to Fahrenheit with F = (C x 9/5) + 32. The program will output two well-organized tables, a Celsius-to-Fahrenheit table as it increases from 0 to 100 degrees, and a Fahrenheit-to-Celsius table as it decreases from 32 to 212 degrees. This exercise not only illustrates the importance of functions in organizing our code for legibility and reusability, it also allows for formatted output to generate aligned tabular displays of data, while also controlling the decimal calculations of the temperature. All in all, this exercise allows the student to learn a real-world formula in programming, the potential usage of loops, and to be able to format the program data into an organized and legible, tabulated format. Overall, it merges both logical and visual elements in C++ programming.

8. Conclusion

By completing these three C++ programs, I gained a better understanding of how functions work and why they are a fundamental part of structuring a program. The main takeaway from all of the tasks was the importance of partitioning problems into smaller and reusable components. The cube volume program introduced me to a simple function to perform mathematics operations, while the hypotenuse and temperature conversions program highlighted the idea that using multiple functions is a way to allow us to develop a program in a modular and organized way. This ultimately helped my understand how function prototypes and function definitions can help with program readability and maintenance.
I also learned how to use input statements and output statements efficiently, how use mathematical formulae within programming, and how to work with real numbers using the double data type. The assignments used other standard C++ libraries like for mathematical operations and for the hope of providing the output formatting functionality. While not necessarily an objective of the activity, learning to use these standard libraries made the task easier by writing clean and professional looking programs that could be both functional and easy to read for users.
In general the activity was a great opportunity to use both logical thinking and coding abilities to develop the solutions to real world problems. Completing the assignments helped me to make connections underpinned some fundamental concepts of programming which included modularity, data handling, and arithmetic computation.