Hands-on Activity 6.2:	
Built-in Functions	
Course Code: CPE 007	**Program:** Computer Engineering
Course Title: Programming Logic and Design	**Date Performed: 10/30/2025**
Section: CPE11S1	**Date Submitted: 10/30/2025**
Name(s): Ramirez, Angel Mae C.	**Instructor: Engr. Jimlord M. Quejado**

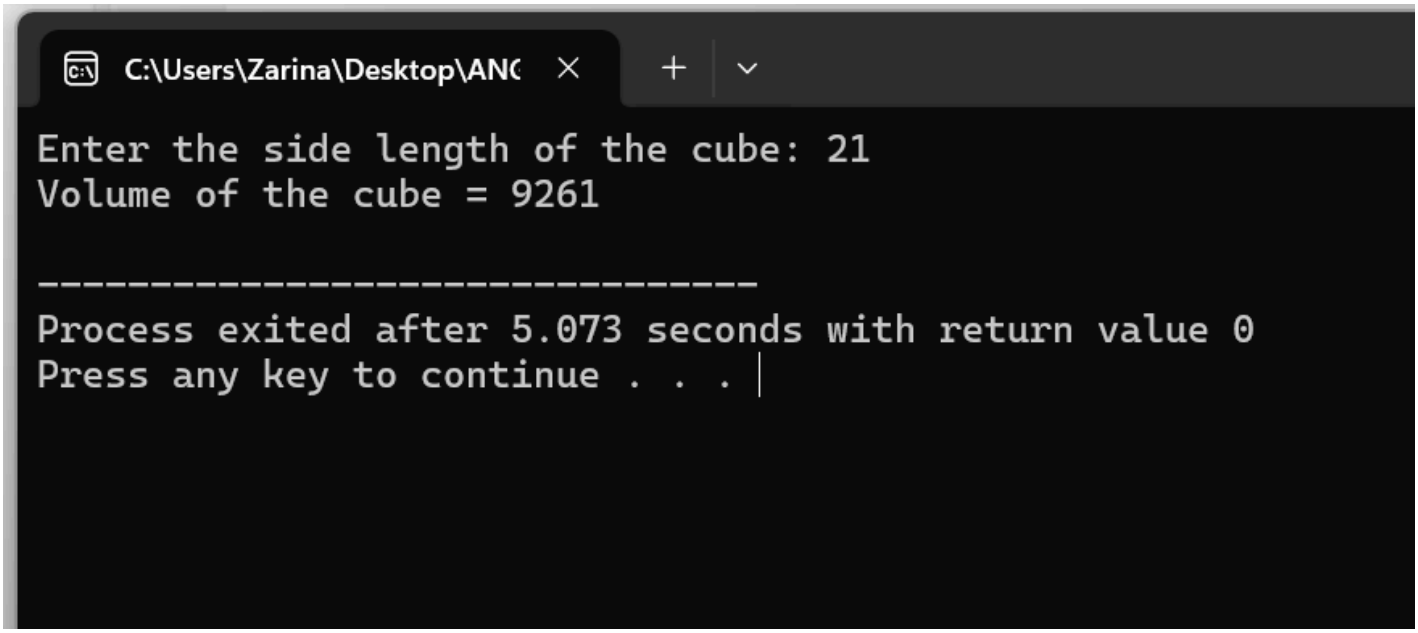
6. Output

1. CODE

built in functions.cpp

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

double volumeCube(double side) {
    return side * side * side;
}

int main() {
    double side;
    cout << "Enter the side length of the cube: ";
    cin >> side;

    cout << "Volume of the cube = " << volumeCube(side) << endl;
    return 0;
}

```

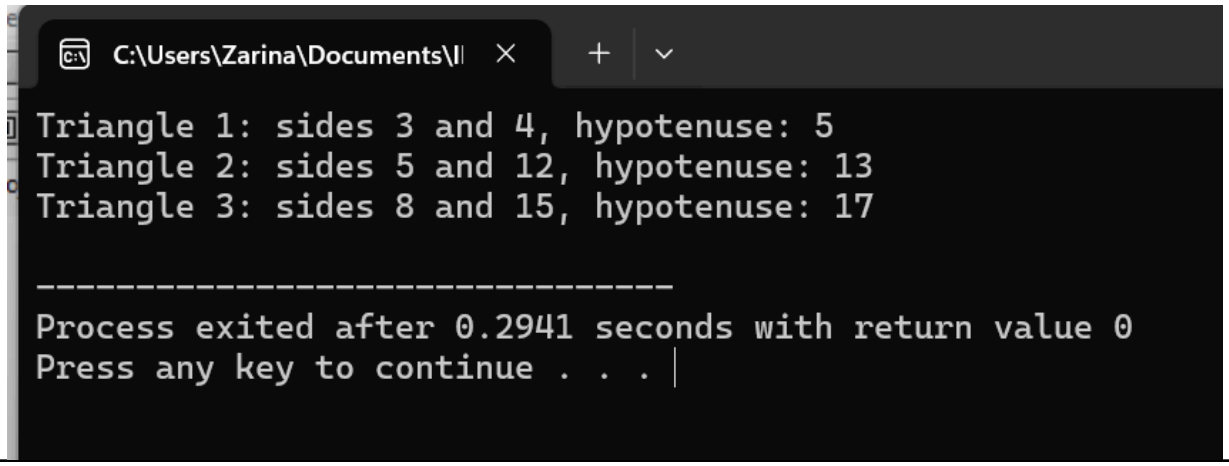
OUTPUT:



2. CODE:

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

// Function prototype
double hypotenuse(double a, double b);

// Function definition
double hypotenuse(double a, double b) {
    return sqrt(a * a + b * b);
}

int main() {
    double side1, side2;

    // Triangle 1
    side1 = 3.0;
    side2 = 4.0;
    cout << "Triangle 1: sides " << side1 << " and " << side2;
    cout << ", hypotenuse: " << hypotenuse(side1, side2) << endl;

    // Triangle 2
    side1 = 5.0;
    side2 = 12.0;
    cout << "Triangle 2: sides " << side1 << " and " << side2;
    cout << ", hypotenuse: " << hypotenuse(side1, side2) << endl;

    // Triangle 3
    side1 = 8.0;
    side2 = 15.0;
    cout << "Triangle 3: sides " << side1 << " and " << side2;
    cout << ", hypotenuse: " << hypotenuse(side1, side2) << endl;

    return 0;
}
```

OUTPUT:



3. CODE

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

// Function prototypes
int celsius(int f);
int fahrenheit(int c);

// Function definitions
int celsius(int f) {
    return round((f - 32) * 5.0 / 9.0);
}

int fahrenheit(int c) {
    return round((c * 9.0 / 5.0) + 32);
}

int main() {
    cout << "Celsius to Fahrenheit:" << endl;
    cout << "C   F   C   F   C   F   C   F   C   F" << endl;

    int count = 0;
    for (int c = 0; c <= 100; c++) {
        cout << setw(3) << c << setw(4) << fahrenheit(c) << " ";
        count++;
        if (count % 10 == 0) cout << endl;
    }

    cout << endl << endl;

    cout << "Fahrenheit to Celsius:" << endl;
    cout << "F   C   F   C   F   C   F   C   F   C" << endl;

    count = 0;
    for (int f = 32; f <= 212; f++) {
        cout << setw(3) << f << setw(4) << celsius(f) << " ";
        count++;
        if (count % 10 == 0) cout << endl;
    }

    cout << endl;
    return 0;
}
```

OUTPUT:

```
C:\Users\Zarina\Documents\F

Celsius to Fahrenheit:
C   F   C   F   C   F   C   F   C   F
  0  32   1  34   2  36   3  37   4  39   5  41   6  43   7  45   8  46   9  48
 10  50  11  52  12  54  13  55  14  57  15  59  16  61  17  63  18  64  19  66
 20  68  21  70  22  72  23  73  24  75  25  77  26  79  27  81  28  82  29  84
 30  86  31  88  32  90  33  91  34  93  35  95  36  97  37  99  38 100  39 102
 40 104  41 106  42 108  43 109  44 111  45 113  46 115  47 117  48 118  49 120
 50 122  51 124  52 126  53 127  54 129  55 131  56 133  57 135  58 136  59 138
 60 140  61 142  62 144  63 145  64 147  65 149  66 151  67 153  68 154  69 156
 70 158  71 160  72 162  73 163  74 165  75 167  76 169  77 171  78 172  79 174
 80 176  81 178  82 180  83 181  84 183  85 185  86 187  87 189  88 190  89 192
 90 194  91 196  92 198  93 199  94 201  95 203  96 205  97 207  98 208  99 210
100 212

Fahrenheit to Celsius:
F   C   F   C   F   C   F   C   F   C
 32   0  33   1  34   1  35   2  36   2  37   3  38   3  39   4  40   4  41   5
 42   6  43   6  44   7  45   7  46   8  47   8  48   9  49   9  50  10  51  11
 52  11  53  12  54  12  55  13  56  13  57  14  58  14  59  15  60  16  61  16
 62  17  63  17  64  18  65  18  66  19  67  19  68  20  69  21  70  21  71  22
 72  22  73  23  74  23  75  24  76  24  77  25  78  26  79  26  80  27  81  27
 82  28  83  28  84  29  85  29  86  30  87  31  88  31  89  32  90  32  91  33
 92  33  93  34  94  34  95  35  96  36  97  36  98  37  99  37 100  38 101  38
102  39 103  39 104  40 105  41 106  41 107  42 108  42 109  43 110  43 111  44
112  44 113  45 114  46 115  46 116  47 117  47 118  48 119  48 120  49 121  49
122  50 123  51 124  51 125  52 126  52 127  53 128  53 129  54 130  54 131  55
132  56 133  56 134  57 135  57 136  58 137  58 138  59 139  59 140  60 141  61
142  61 143  62 144  62 145  63 146  63 147  64 148  64 149  65 150  66 151  66
152  67 153  67 154  68 155  68 156  69 157  69 158  70 159  71 160  71 161  72
162  72 163  73 164  73 165  74 166  74 167  75 168  76 169  76 170  77 171  77
172  78 173  78 174  79 175  79 176  80 177  81 178  81 179  82 180  82 181  83
182  83 183  84 184  84 185  85 186  86 187  86 188  87 189  87 190  88 191  88
192  89 193  89 194  90 195  91 196  91 197  92 198  92 199  93 200  93 201  94
```

7. Supplementary Activity

1.ANALYSIS

in this program that I made it is about finding the volume of a cube using a function. In this code, I first included the header file iostream so I can use input and output commands like cin and cout. Then, I created a function called volumeCube() which takes one parameter named side. Inside that function, I just returned the formula for the cube's volume, which is side × side × side. I used double as the data type because it allows decimal values, so it can handle sides that aren't whole numbers. In the main function, I declared a variable named side to store the user's input. The program asks the user to enter the side length of the cube using cout, and then it reads the value with cin. After that, the program calls the volumeCube() function and displays the result on the screen. The output shows the message Volume of the cube followed by the computed value.

2.ANALYSIS

This C++ program that I made is used to find the hypotenuse of a right triangle using a function. I started by including the header files <iostream> for input and output, and <cmath> because I needed the sqrt() function to calculate the square root. Then, I created a function called hypotenuse() that takes two parameters, a and b, which represent the two sides of a right triangle. Inside that function, I returned the result of the Pythagorean theorem formula: $\sqrt{(a^2 + b^2)}$. In the main() function, I declared two variables side1 and side2 to store the triangle sides. I made three examples of triangles with different side lengths (3,4), (5,12), and (8,15). For each triangle, the program prints the sides and then calls the

hypotenuse() function to calculate and display the hypotenuse. The output shows the side lengths followed by the computed hypotenuse value for each triangle.

3.ANALYSIS

This C++ program that I made is about converting temperatures between Celsius and Fahrenheit using functions. At the start, I included three header files iostream for input and output, iomanip for formatting with setw, and cmath for rounding numbers using round. Then, I created two functions thw celsius and the fahrenheit. The celsius function converts Fahrenheit to Celsius using the formula (f - 32) * 5 / 9, while the fahrenheit() function converts Celsius to Fahrenheit using (c * 9 / 5) + 32. Both results are rounded to the nearest integer for cleaner output. In the main function, the program first prints a table that shows the conversion of Celsius to Fahrenheit from 0°C to 100°C. I used a for loop and the setw function to make the output neatly aligned in columns. After that, another loop displays the Fahrenheit to Celsius conversion from 32°F to 212°F. I also used a counter to print a new line every 10 values to make the table look more organized. While making this program, I admit I struggled a little with the formatting and the use of setw because at first, my output didn't align properly. I also had to review how the rounding function works in cmath. But after testing and adjusting the code, I finally got it right.

8. Conclusion

While doing this activity, I learned about how functions can make a program easier to understand and organize. Before, I didn't really see the importance of separating tasks into different functions, but now I get it. By using the `celsius()` and fahrenheit functions, the code became more readable and less confusing. I also realized that using libraries like `<iomanip>` and `<cmath>` can make outputs more accurate and presentable, which really helped improve how the program looks and works.

While I was doing the program, I honestly had a hard time getting the spacing and alignment right. The `setw()` part was kind of tricky at first because the numbers wouldn't line up properly. I also messed up the formulas a few times, but after testing and checking, I finally understood how each part of the conversion works. It took some patience, but it felt good seeing the program finally display the correct results in a clean table format.

Overall, this task helped me improve my logic and patience when coding. I realized that programming isn't just about getting the answer, it's about making the code clear, correct, and easy to read. Even though I struggled a bit, it was worth it because I now understand how functions and loops work together to solve real problems. This exercise made me more confident in using C++ and more interested in exploring what else I can do with it.