

Activity No. 6.1

Hands-on Activity 6.2: Hands-on Activity 6.1: Built-in Functions

Course Code: CPE007

Program: Computer Engineering

Course Title: Programming Logic and Design

Date Performed: October 29, 2025

Section: CPE11S1

Date Submitted: October 30, 2025

Name(s): Tobias, Lawrence C.

Instructor: Engr. Jimlord M. Quejado

6. Output

1.

WHAT.cpp

```
1  #include <iostream>
2  using namespace std;
3
4  double volume(double s);
5  double volume(double s) {
6      return s * s * s;
7  }
8  int main() {
9      double side;
10     cout << "Enter the side length of the cube: ";
11     cin >> side;
12     double vol = volume(side);
13     cout << "The volume of the cube is: " << vol << endl;
14     return 0;
15 }
```

C:\Dev-Cpp\WHAT.exe

```
Enter the side length of the cube: 13
The volume of the cube is: 2197
```

```
-----
Process exited after 2.735 seconds with return value 0
Press any key to continue . . . |
```

2.

WHAT.cpp DO YOU MEAN.cpp

```
1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  double hypotenuse(double a, double b);
6  double hypotenuse(double a, double b) {
7      return sqrt(a * a + b * b);
8  }
9
10 int main() {
11     double side1, side2;
12
13     side1 = 3.0;
14     side2 = 4.0;
15     cout << "Triangle 1: sides " << side1 << " and " << side2
16          << ", hypotenuse: " << hypotenuse(side1, side2) << endl;
17
18     side1 = 5.0;
19     side2 = 12.0;
20     cout << "Triangle 2: sides " << side1 << " and " << side2
21          << ", hypotenuse: " << hypotenuse(side1, side2) << endl;
22
23     side1 = 8.0;
24     side2 = 15.0;
25     cout << "Triangle 3: sides " << side1 << " and " << side2
26          << ", hypotenuse: " << hypotenuse(side1, side2) << endl;
27
28     return 0;
29 }
30
```

C:\Dev-Cpp\DO YOU MEAN.e: × + ▾

```
Triangle 1: sides 3 and 4, hypotenuse: 5
Triangle 2: sides 5 and 12, hypotenuse: 13
Triangle 3: sides 8 and 15, hypotenuse: 17
```

```
-----
Process exited after 0.3152 seconds with return value 0
Press any key to continue . . .
```

3.

WHAT.cpp DO YOU MEAN.cpp BY THAT.cpp

```

1  #include <iostream>
2  #include <iomanip>
3  #include <cmath>
4  using namespace std;
5
6  int celsius(int f);
7  int fahrenheit(int c);
8
9  int celsius(int f) {
10     return round((f - 32) * 5.0 / 9.0);
11 }
12
13 int fahrenheit(int c) {
14     return round((c * 9.0 / 5.0) + 32);
15 }
16
17 int main() {
18     cout << "Celsius to Fahrenheit:" << endl;
19     cout << "C  F  C  F  C  F  C  F  C  F  C  F  C  F  C  F  C  F" << endl;
20     int count = 0;
21     for (int c = 0; c <= 100; c++) {
22         cout << setw(3) << c << setw(4) << fahrenheit(c) << " ";
23         count++;
24         if (count % 10 == 0) cout << endl;
25     }
26     cout << endl;
27
28     cout << "Fahrenheit to Celsius:" << endl;
29     cout << "F  C  F  C  F  C  F  C  F  C  F  C  F  C  F  C  F  C" << endl;
30     count = 0;
31     for (int f = 32; f <= 212; f++) {
32         cout << setw(3) << f << setw(4) << celsius(f) << " ";
33         count++;
34         if (count % 10 == 0) cout << endl;
35     }
36     cout << endl;
37
38     return 0;
39 }
40

```

Celsius to Fahrenheit:

C	F	C	F	C	F	C	F	C	F	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	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
202	94	203	95	204	96	205	96	206	97	207	97	208	98	209	98	210	99	211	99
212	100																		

 Process exited after 0.345 seconds with return value 0
 Press any key to continue . . . |

7. Supplementary Activity

Analysis:

1. For this program, I made a function called volume(double s) and I declared it at the top so the program knows it exists. This function returns $s * s * s$, which is the formula for the volume of a cube. In the main part, I declared a variable named side to store the value that the user will input. After the user types the side length, I called the volume function and stored its result in another variable named vol. Then the program displays the volume that was calculated. So basically, the program asks for the cube's side, calculates $side^3$ using the function I made, and shows the answer. When I tried it with the side length of 13, the output was: 13 The volume of the cube is: 2197.

2. For this program, I created a function called `hypotenuse(double a, double b)` and declared it at the top so the program knows it will be used. This function returns `sqrt(a * a + b * b)`, which follows the Pythagorean Theorem for finding the hypotenuse of a right triangle. In the main part of the program, I declared two variables named `side1` and `side2`, and I assigned different values to them three times to represent three different triangles. For each pair of side lengths, I called the `hypotenuse` function to compute and show the length of the hypotenuse. So basically, I just changed the side values for each triangle and used the same function to calculate their hypotenuses. When the program ran, the output was: Triangle 1 with sides 3 and 4 had a hypotenuse of 5, Triangle 2 with sides 5 and 12 had a hypotenuse of 13, and Triangle 3 with sides 8 and 15 had a hypotenuse of 17.
3. For this program, I created two functions: `celsius(int f)` and `fahrenheit(int c)`, and I declared them at the top so that the program knows they will be used. The `celsius` function converts a Fahrenheit value into Celsius using the conversion formula, and the `fahrenheit` function converts a Celsius value into Fahrenheit. Both conversions use `round` to make sure the results are whole numbers. In the main part of the program, I used two loops to generate conversion tables. The first loop goes from 0 to 100 to convert Celsius values to Fahrenheit, and the second loop goes from 32 to 212 to convert Fahrenheit values to Celsius. I also used a counter to help format the output into neat columns. So in the end, the program prints two organized tables: one showing Celsius to Fahrenheit conversions and another showing Fahrenheit to Celsius conversions.

8. Conclusion

In this activity, I made programs that each had their own task. One part was about finding the volume of a cube, another part focused on solving the hypotenuse of a triangle, and the last part was about converting temperatures between Celsius and Fahrenheit. I made sure that each part worked correctly by checking if the answers matched the formulas. Sometimes I needed to look at the numbers again to make sure the calculation was right, but once everything worked, it was nice to see the correct results show up. Overall, this activity helped me organize my code better and understand how breaking a program into clear steps makes it easier to read, use, and understand.