

Documentation of Chapter 1

Md. Mahfuj Hasan Shohug

BDCOM0019

Exercise 1-1:

Let's, run the "Hello, world!" program on my system. And now experiment with leaving out parts of the program, to see the error messages and write down below.

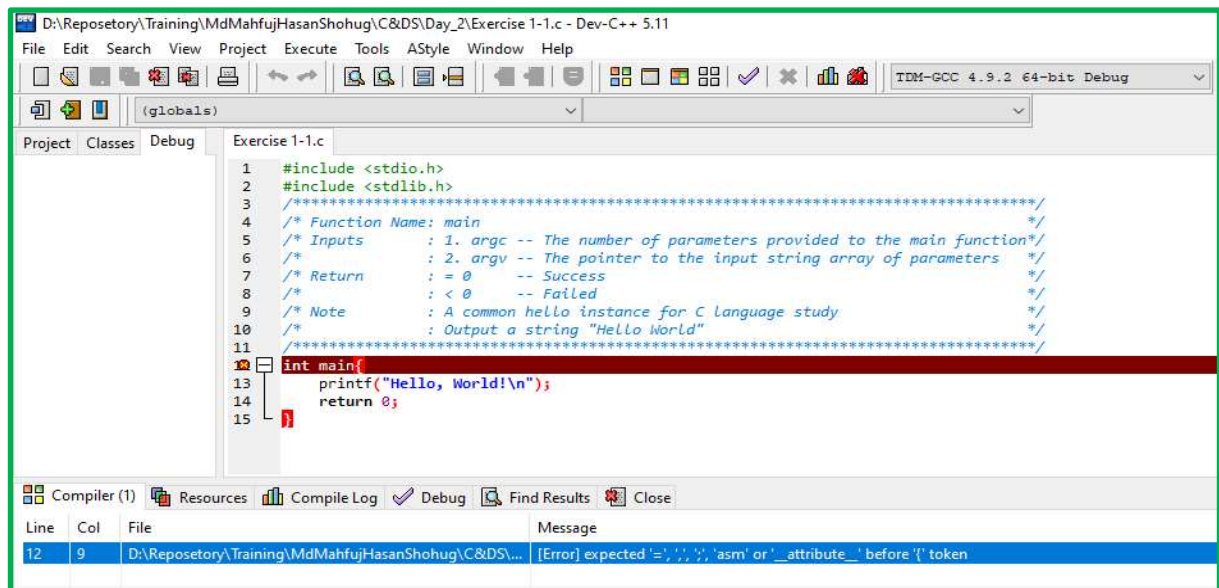


Fig-1: A function must need **parenthesis**, here I got syntax errors for not typing the parenthesis and the program not executed.

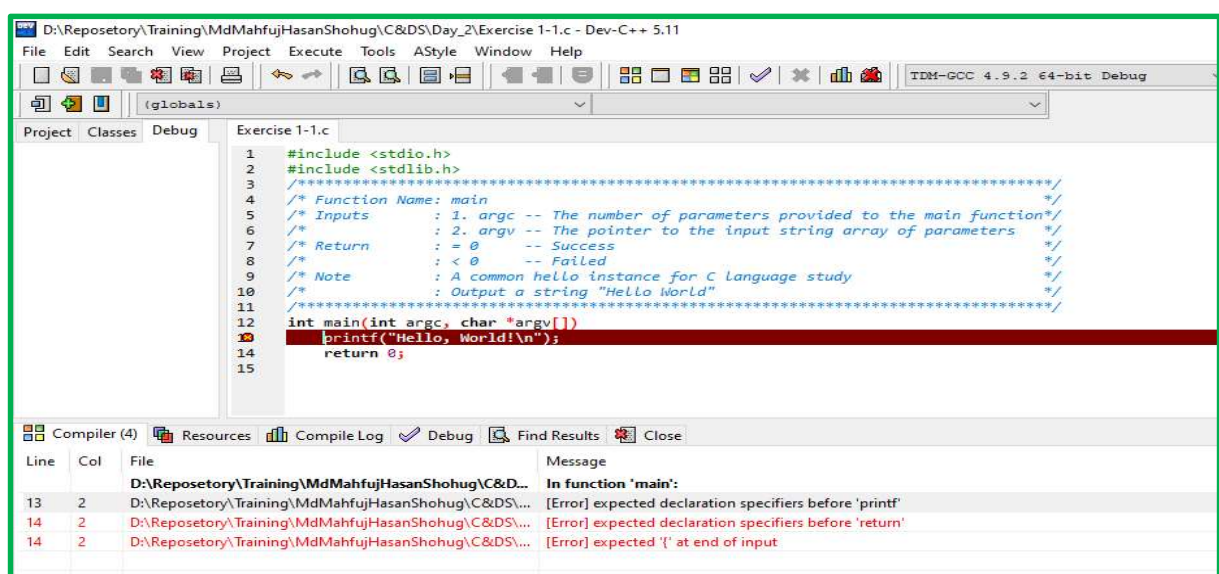


Fig-2: The bodies of all function must need to use **Braces**, this is also a syntax error therefor program not executed.

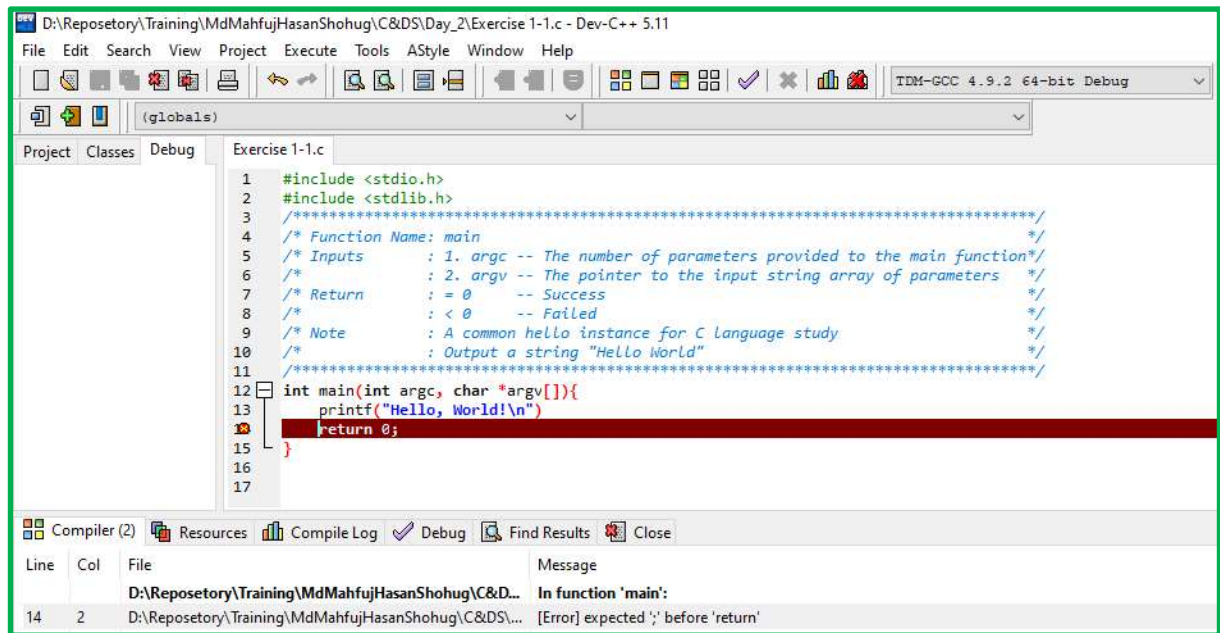


Fig-3: Must need to use semicolon for execute the program. This is also a syntax error all statements must end with a semicolon.

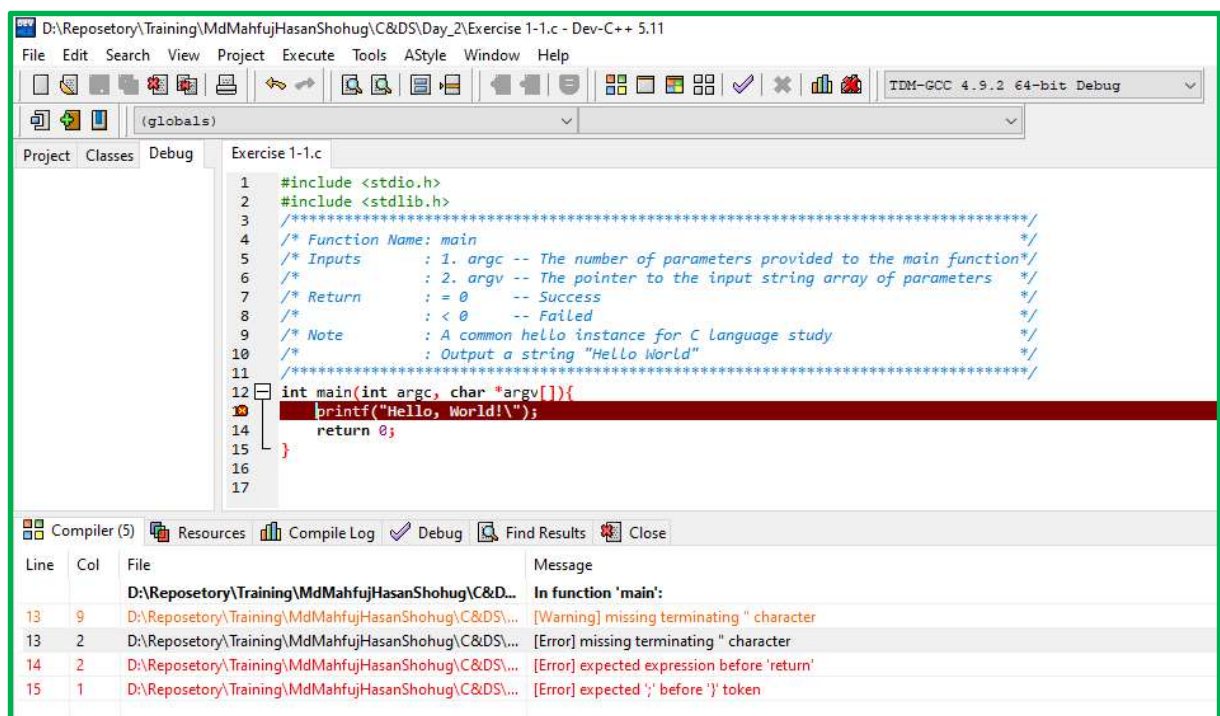


Fig-4: Notice that \n represents only a single character here I am using only “/” that is not extensible mechanism for representing hard-to-type or invisible characters.

Debug: Here I am debugging the two times to show the debug message on there. First I am write the code without any kind of return value here is my debugging step which one I done:

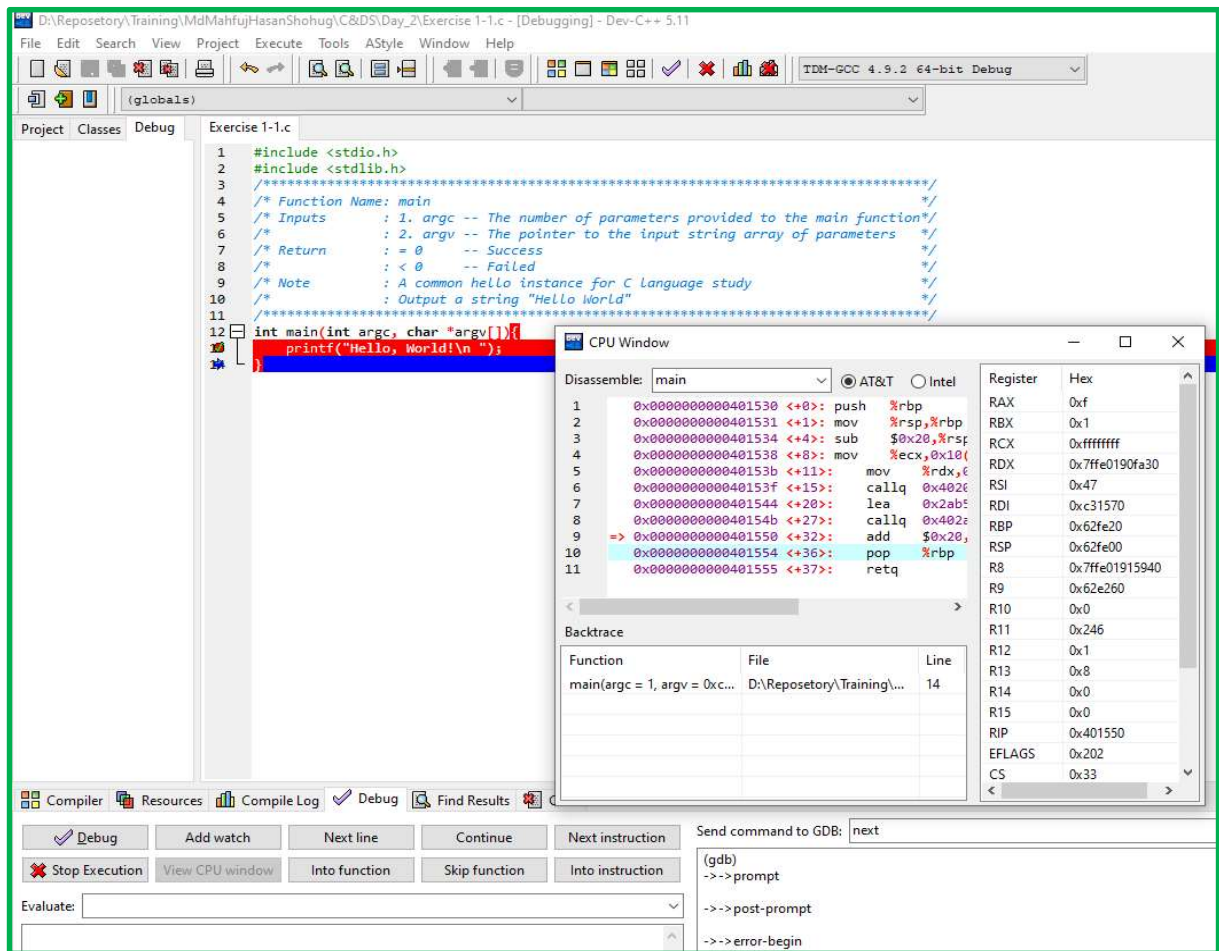


Fig-5: Debug without return.
Debug with return value 0.

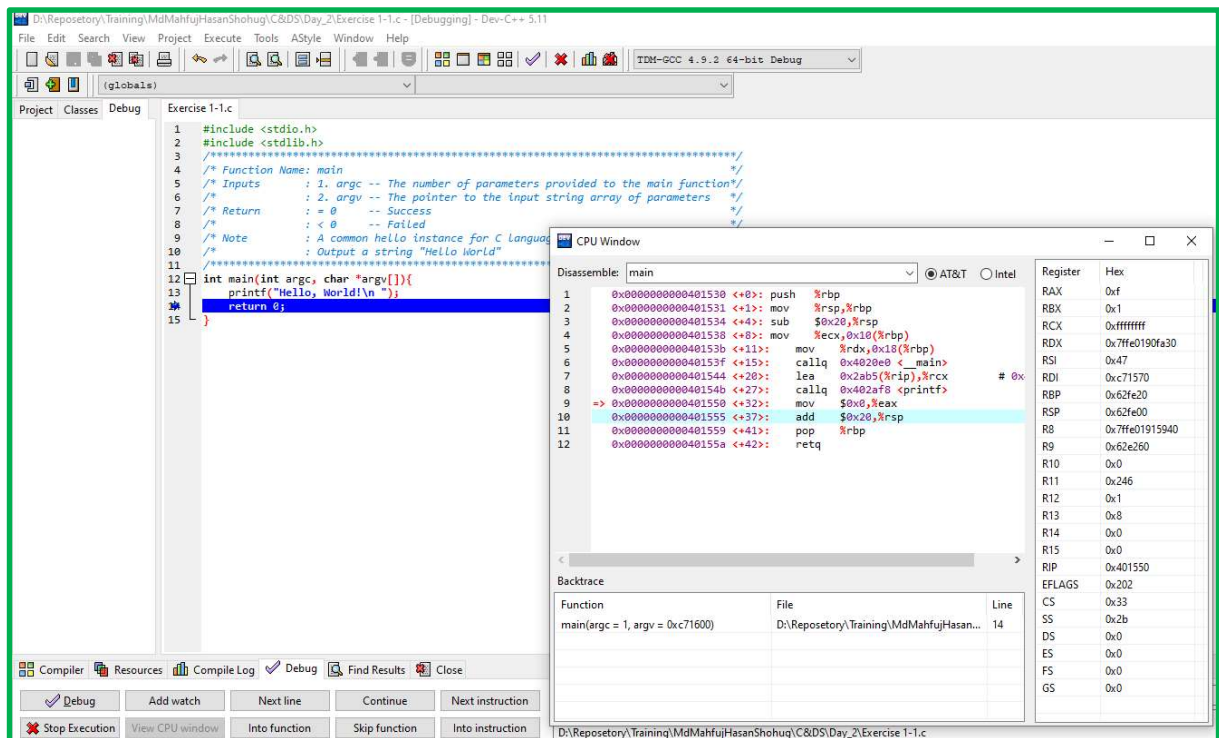


Fig-6: Debug the code and execute the whole program.

Exercise 1-2:

Experiment to find out what happens when prints argument string contains `\c` and also using some special character. Those are the “escape sequence” here I want to showing the other escape sequence those are also usable.

- `\C` Output: Program execute only C with the warning message that is unknown escape sequence, the “`\`” will not be print.

Coding part:

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  /*****
4  /* Function Name: main
5  /* Inputs      : 1. argc -- The number of parameters provided to the main function*/
6  /*             : 2. argv -- The pointer to the input string array of parameters */
7  /* Return      : = 0 -- Success
8  /*             : < 0 -- Failed
9  /* Note       : A common hello instance for C language study
10 /*            : Output a string "Hello World"
11 /*****/
12 int main(int argc, char *argv[])
13 {
14     printf("Hello, \c World!");
15     return 0;
16 }

```

Compiler (2) Resources Compile Log Debug Find Results Close

Line	Col	File	Message
13	9	D:\Repository\Training\MdMahfujHasanShohug\C&D...	In function 'main': [Warning] unknown escape sequence: '\c'

Output:

```

D:\Repository\Training\MdMahfujHasanShohug\C&D\Day_2\Exercise 1-2.exe
Hello, c World!
-----
Process exited after 0.03058 seconds with return value 0
Press any key to continue . . .

```

There are many escape sequence have on the C language. Here I am referring some of escape sequence as an example:

- \a: It's for the audible bell. When program will be run the one audible sound will be listen.

Coding part and Output: Show output with a notification sound.

The screenshot shows the Dev-C++ IDE with the following code in the editor:

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 /* Function Name: main
5 /* Inputs      : 1. argc -- The number of parameters provided to the main function*/
6 /*            : 2. argv -- The pointer to the input string array of parameters */
7 /* Return      : = 0 -- Success
8 /*            : < 0 -- Failed
9 /* Note       : A common hello instance for C language study
10 /*           : Output a string "Hello World"
11
12 int main(int argc, char *argv[])
13 {
14     printf("Hello, World!\a");
15     return 0;
16 }

```

The output window shows the following text:

```

Hello, World!
-----
Process exited after 0.02684 seconds with return value 0
Press any key to continue . . .

```

- \r: It's for Carriage Return. This escape sequence works to position the cursor at the beginning of the line.

Coding part and output: Here showing “World” because \r bring the last word on the beginning of the line

The screenshot shows the Dev-C++ IDE with the following code in the editor:

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 /* Function Name: main
5 /* Inputs      : 1. argc -- The number of parameters provided to the main function*/
6 /*            : 2. argv -- The pointer to the input string array of parameters */
7 /* Return      : = 0 -- Success
8 /*            : < 0 -- Failed
9 /* Note       : A common hello instance for C language study
10 /*           : Output a string "Hello World"
11
12 int main(int argc, char *argv[])
13 {
14     printf("Hello,\r World!");
15     return 0;
16 }

```

The output window shows the following text:

```

World!
-----
Process exited after 0.02878 seconds with return value 0
Press any key to continue . . .

```

Exercise 1.3:

Modifying the temperature conversion program to print a heading from the given table (Fahrenheit to Celsius). Code file:

The screenshot shows a C++ IDE with the source code for Exercise 1-3.c and its compilation output.

Source Code (Exercise 1-3.c):

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  /*****
4  ** Function Name: main
5  ** Inputs      : 1. argc -- The number of parameters provided to the main function**
6  **             : 2. argv -- The pointer to the input string array of parameters **
7  ** Variable    : temp_fahr -- The fahrenheit tempture value
8  **             : temp_celsi-- The Celsius tempture value
9  ** Return      : = 0 -- Success
10 **            : < 0 -- Failed
11 ** Note       : Modifying the temperature conversion program table
12 *****/
13 int main(int argc, char *argv[])
14 {
15     float temp_fahr, temp_celsi; // declire the float variable
16     int low_temp = 0, high_temp = 300, step_num = 20; // initialize the value
17     printf("Fahrenheit \t Celsius\n"); // define table header
18     printf("----- \t ----- \n");
19
20     temp_fahr = low_temp;
21     while(temp_fahr <= high_temp)
22     {
23         temp_celsi = 5*(temp_fahr-32)/9; // fahrenheit to celsius formula
24         printf("%.2f \t %.2f\n", temp_fahr, temp_celsi);
25         temp_fahr = temp_fahr + step_num;
26     }
27     return 0;
28 }

```

Compilation Output:

```

Compiling single file...
- Filename: D:\Reposetory\Training\MdMahfujHasanShohug\C&DS\Day_2\Exercise 1-3.c
- Compiler Name: TDM-GCC 4.9.2 64-bit Debug

Processing C source file...
- C Compiler: C:\Program Files (x86)\Dev-Cpp\MinGW64\bin\gcc.exe
- Command: gcc.exe "D:\Reposetory\Training\MdMahfujHasanShohug\C&DS\Day_2\Exercise 1-3.c" -o "D:\
Compilation results...
- Errors: 0
- Warnings: 0
- Output Filename: D:\Reposetory\Training\MdMahfujHasanShohug\C&DS\Day_2\Exercise 1-3.exe
- Output Size: 154.3369140625 KiB
- Compilation Time: 0.19s

```

Output:

The screenshot shows the output of the program, which is a table of Fahrenheit to Celsius conversions. The output is as follows:

```

D:\Reposetory\Training\MdMahfujHasanShohug\C&DS\Day_2\Exercise 1-3.exe
Fahrenheit      Celsius
-----
0.00            -17.78
20.00           -6.67
40.00           4.44
60.00           15.56
80.00           26.67
100.00          37.78
120.00          48.89
140.00          60.00
160.00          71.11
180.00          82.22
200.00          93.33
220.00          104.44
240.00          115.56
260.00          126.67
280.00          137.78
300.00          148.89
-----
Process exited after 0.03147 seconds with return value 0
Press any key to continue . . .

```

Exercise 1.4:

Now here I am writing a program to print the corresponding Celsius to Fahrenheit table.

Coding file:

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  /*****
4  ** Function Name: main
5  ** Inputs      : 1. argc -- The number of parameters provided to the main function**
6  **             : 2. argv -- The pointer to the input string array of parameters **
7  ** Variable    : temp_fahr -- The fahrenheit tempture value
8  **             : temp_celsi-- The Celsius tempture value
9  ** Return      : = 0 -- Success
10 **            : < 0 -- Failed
11 ** Note       : temperature conversion celsius to fahrenheit
12 *****/
13 int main(int argc, char *argv[])
14 {
15     float temp_fahr, temp_celsi; // declire the float variable for tempture
16     int low_temp = 0, high_temp = 300, step_num = 20; // initialize the value
17     printf("Celsius \t Fahrenheit\n"); // define table header
18     printf("----- \t ----- \n");
19
20     temp_celsi = low_temp;
21     while(temp_celsi <= high_temp)
22     {
23         temp_fahr = (1.8*temp_celsi)+32; // celsius to fahrenheit formula
24         printf("%.2f \t \t %.2f\n", temp_celsi, temp_fahr);
25         temp_celsi = temp_celsi + step_num;
26     }
27     return 0;
28 }
  
```

Compiler: Resources Compile Log Debug Find Results Close

Compiling single file...

Filename: D:\Reposetory\Training\MdMahfujHasanShohug\C&DS\Day_2\Exercise 1-4.c
Compiler Name: TDM-GCC 4.9.2 64-bit Debug

Processing C source file...

C Compiler: C:\Program Files (x86)\Dev-Cpp\MinGW64\bin\gcc.exe
Command: gcc.exe "D:\Reposetory\Training\MdMahfujHasanShohug\C&DS\Day_2\Exercise 1-4.c" -o "D:\Reposetory\Training\MdMahfujHasanShohug\C&DS\Day_2\Exercise 1-4.exe"

Compilation results...

Errors: 0
Warnings: 0
Output Filename: D:\Reposetory\Training\MdMahfujHasanShohug\C&DS\Day_2\Exercise 1-4.exe
Output Size: 154.3369140625 KiB
Compilation Time: 0.19s

Output:

```

D:\Reposetory\Training\MdMahfujHasanShohug\C&DS\Day_2\Exercise 1-4.exe
Celsius      Fahrenheit
-----
0.00         32.00
20.00        68.00
40.00        104.00
60.00        140.00
80.00        176.00
100.00       212.00
120.00       248.00
140.00       284.00
160.00       320.00
180.00       356.00
200.00       392.00
220.00       428.00
240.00       464.00
260.00       500.00
280.00       536.00
300.00       572.00
-----
Process exited after 0.02988 seconds with return value 0
Press any key to continue . . .
  
```