

Learn CODE: Start with C

Article is prepared by –

Sorowar Mahabub, Studies B.Sc in CSE,

Int' Islamic University Chittagong (IIUC)

Cell: 01834756433, 01521564157(whasApp available)

Email : sorowarmahabub1709vip@gmail.com

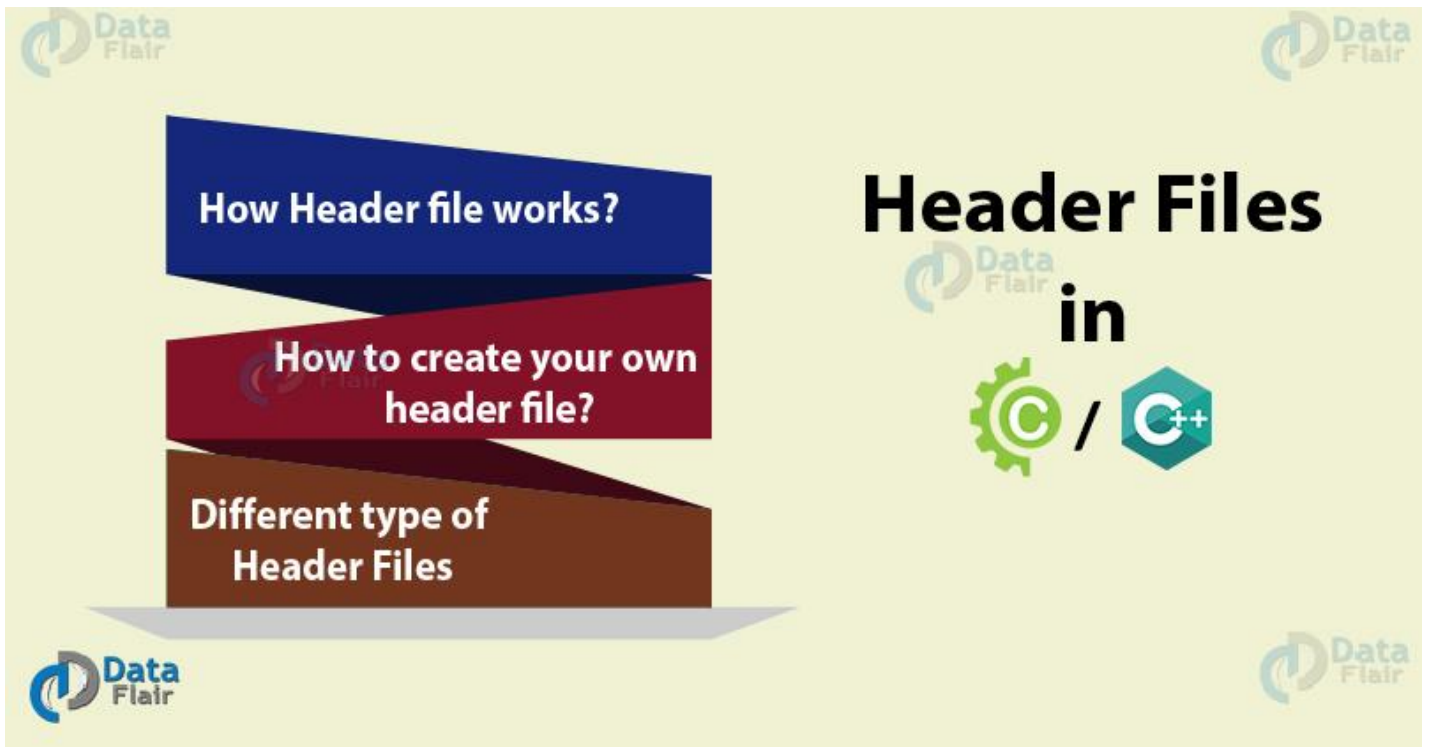
Article Reference : Google, DataFlair_blog

Header Files in C/C++ (Create Header Files Within Seconds)

Are you aware of the various Header Files in C/C++? If not, there's no need to worry. We are going to discuss each and everything about header files in C/C++. Each program requires at least one header file to work.

Before we begin our discussion, it is important to understand, what are preprocessor directives? These are the basic building blocks of header files in C and C++. The term “**preprocessor**” is *self-explanatory*. The word “pre” means “before” and the word “processor” means “to make something”. Before the source code is compiled, it gets automatically processed due to the presence of preprocessor directives.

So, let's start and explore the depth of header files of C/C++, which will be helping you out in numerous ways.



1. What is a Header File in C and C++?

The [C/C++ Standard Library](#) offers its users a variety of functions, one of which is header files.

In C++, all the header files may or may not end with the .h extension but in C, all the header files must necessarily begin with the.h extension.

A header file in C/C++ contains:

- Function definitions
- Data type definitions
- Macros

Header files offer these features by importing them into your program with the help of a preprocessor directive called **#include**. These preprocessor directives are responsible for instructing the C/C++ compiler that these files need to be processed before compilation.

Every C program should necessarily contain the header file **<stdio.h>** which stands for standard input and output used to take input with the help of **scanf() function** and display the output using **printf() function**.

C++ program should necessarily contain the header file **<iostream>** which stands for input and output stream used to take input with the help of “**cin>>**” **function** and display the output using “**cout<<**” **function**.

From this example, it is clear that each header file of C and C++ has its own specific function associated with it.

Basically, header files are of 2 types:

1. **Standard library header files:** These are the pre-existing header files already available in the C/C++ compiler.
2. **User-defined header files:** Header files starting #define can be designed by the user.

2. Syntax of Header File in C/C++

We can define the syntax of the header file in 2 ways:

- **#include<filename.h>**

The name of the header file is enclosed within angular brackets. It is the most common way of defining a header file. As discussed earlier, in C, all header files would compulsorily begin with the .h extension otherwise, you would get a compilation error but it is not the case in C++.

For example,

```
#include<string.h> // Supported both in C and C++
```

And,

```
#include<vector> // An exclusive feature of C++
```

- **#include“filename.h” or #include “filename”**

The name of the header file is enclosed within double-quotes. It is usually preferred when defining user-defined header files.

For example,

```
#include"stdlib.h" // Available in both C and C++
```

And,

```
#include"iostream" // Exclusive to C++
```

Key takeaway: We cannot include the same header file in the same program twice.

Are you aware of the [Rules of Syntax in C Programming](#)

3. How Header Files Work?

The source file contains #include which is responsible for directing the C/C++ compiler that this file needs to be processed before compilation and includes all the necessary data type and function definitions.

4. How to Create your own Header File in C/C++?

Instead of writing a large and complex code, you can create your own header files and include it in the C/C++ library to use it whenever you wish as frequently as you like. It enhances code functionality and readability.

Let us understand how to create your own header file in C++ with the help of an example.

Consider a problem where you want to compute the factorial of a number. Since it not pre-defined in the standard C++ library, you can create it by yourself!

The Steps involved are-

Step – 1

Write your own code in C++ and save the file with a .h extension instead of a .cpp, because you are creating a header file, not a C++ program. The name of the file you save with **.h extension** would be the name of your header file. Suppose you named it **factorial.h**.

```
1.int factorial(int number)
2.{
3.int iteration, factorial=1;
4.for(iteration=1; iteration<=number; iteration++)
5.{
6.factorial=factorial*iteration;
7.}
8.return factorial;
9.}
```

Step – 2

Open a fresh window and include your header file. In this case, you can write in two ways:

#include“factorial.h” – Enclosing the header file name within double quotes signifies that the header file of C and C++ is located in the present folder you are working with. It is a preferred practice to include user-defined header files in this manner.

#include<factorial.h> – Enclosing the header file name within angular brackets signifies that the header file is located in the standard folder of all other header files of C/C++.

Step – 3

After the code is written using your file with the .h extension, compile and run your program. This is a C++ program to find the factorial of a number using a self-created header file:

```
1. #include <iostream>
2. #include"factorial.h"
3. using namespace std;
4.
5. int main()
6. {
7.
8. cout<<"Welcome to DataFlair tutorials!"<<endl<<endl;
9.
10. int positive_integer;
11. cout<<"Enter a positive integer: "<<endl;
12. cin>>positive_integer;
13. cout<<"The factorial of " << positive_integer << " is: " << factorial(positive_integer) <<endl;
14.
15. return 0;
16. }
```

Key takeaway: The header file and your C/C++ program should be in the same folder.

Different Types of C/C++ Header File

Do you ever think, how many header files are there in C/C++ Programming Language?

There are many header files present in C and C++. Even we can create them according to our requirement. In order to access the Standard Library functions, certain header files in C/C++ need to be included before writing the body of the program.

C/C++ Header File

Let's have a look at these Header files in C and C++:

1. **#include<stdio.h> (Standard input-output header)**

Used to perform input and output operations in C like scanf() and printf().

2. **#include<string.h> (String header)**

Perform string manipulation operations like strlen and strcpy.

3. **#include<conio.h> (Console input-output header)**

Perform console input and console output operations like clrscr() to clear the screen and getch() to get the character from the keyboard.

4. **#include<stdlib.h> (Standard library header)**

Perform standard utility functions like dynamic memory allocation, using functions such as malloc() and calloc().

5. **#include<math.h> (Math header)**

Perform mathematical operations like sqrt() and pow(). To obtain the square root and the power of a number respectively.

6. **#include<ctype.h>(Character type header)**

Perform character type functions like isaplha() and isdigit(). To find whether the given character is an alphabet or a digit respectively.

7. **#include<time.h>(Time header)**

Perform functions related to date and time like `setdate()` and `getdate()`. To modify the system date and get the CPU time respectively.

8. `#include<assert.h>` (Assertion header)

It is used in program assertion functions like `assert()`. To get an integer *data type in C/C++* as a parameter which prints `stderr` only if the parameter passed is 0.

9. `#include<locale.h>` (Localization header)

Perform localization functions like `setlocale()` and `localeconv()`. To set locale and get locale conventions respectively.

10. `#include<signal.h>` (Signal header)

Perform signal handling functions like `signal()` and `raise()`. To install signal handler and to raise the signal in the program respectively

11. `#include<setjmp.h>` (Jump header)

Perform jump functions.

12. `#include<stdarg.h>` (Standard argument header)

Perform standard argument functions like `va_start` and `va_arg()`. To indicate start of the variable-length argument list and to fetch the arguments from the variable-length argument list in the program respectively.

13. `#include<errno.h>` (Error handling header)

Used to perform error handling operations like `errno()`. To indicate errors in the program by initially assigning the value of this function to 0 and then later changing it to indicate errors.

List of C++ Header File

Following are some C++ header files which are not supported in C-

1. **#include<iostream> (Input Output Stream)** – Used as a stream of Input and Output.
2. **#include<iomanip.h> (Input-Output Manipulation)** – Used to access set() and setprecision().
3. **#include<fstream.h> (File stream)** – Used to control the data to read from a file as an input and data to write into the file as an output.

Summary

A C/C++ programmer should be well acquainted with the use of header files. In this tutorial, we got to know about the various header files available in C/C++, how they are defined, how they work and how to create a user-defined header file. We ended our discussion by summarizing the function of various kinds of header files

Bismillah Hir Rahmanir Rahim

"Let's appear in thy smiling"



Md. Sorowar Mahabub Rabby | 01834-756433 01756-043655

*Studies B. Sc in CSE(Computer Science & Engineering),
Int' Islamic University Chittagong (IIUC).*

Baroierhat, Mirsharai, Chittagong.

Sorowar Mahabub Sorowar Mahabub
sorowarmahabub1709vip@gmail.com