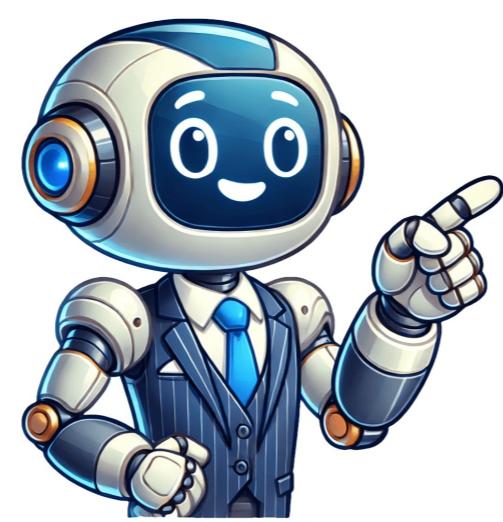


**Click to prove
you're human**



Functions may be return type functions and non-return type functions. The non-return type functions do not return any value to the calling function; the type of such functions is void. These functions may or may not have any argument to act upon. A few illustrations of such functions are given below.

```
void Write (void) { printf("You need a compiler for learning C language."); }
```

The first line in the above definition may also be written as void Write () { // Definition of Display printf("Play an outdoor game for better health."); } The expected output of the above program is as given below.

A function is a block of code which only runs when it is called. You can pass data, known as parameters, into a function. Functions are used to perform certain actions, and they are important for reusing code. Define the code once, and use it many times. Predefined Functions So it turns out you already know what a function is. You have been using it the whole time while studying this tutorial! For example, main() is a function, which is used to execute code, and printf() is a function; used to output/print text to the screen: int main() { printf("Hello World!"); return 0;} Try it Yourself Create a Function To create (often referred to as declare) your own function, specify the name of the function, followed by parentheses () and curly brackets {}:

```
void myFunction() { // code to be executed }
```

Example Explained myFunction() is the name of the function void means that the function does not have a return value. You will learn more about return values later in the next chapter inside the function (the body). add() code that defines what the function should do Call a Function Declared functions are not executed immediately. They are "saved for later use", and will be executed when they are called. To call a function, write the function's name followed by two parentheses () and a semicolon. In the following example, myFunction() is used to print a text (the action), when it is called: Inside main, call myFunction(); // Create a function void myFunction() { printf("I just got executed!"); }

executed!"}; int main() { myFunction(); // call the function return 0;} // Outputs "I just got executed!" Try it Yourself A function can be called multiple times: void myFunction() { printf("I just got executed!"); } int main() { myFunction(); myFunction(); return 0;} // I just got executed!// I just got executed!// I just got executed!

Calculate the Sum of Numbers You can put almost whatever you want inside a function. The purpose of the function is to save the code, and execute it when you need it. Like in the example below, we have created a function to calculate the sum of two numbers. Whenever you are ready to execute the function (and perform the calculation), you just call it: void calculateSum() { int x = 5; int y = 10; int sum = x + y; printf("The sum of x + y is: %d", sum); } int main() { calculateSum(); // call the function return 0;} // Outputs The sum of x + y is: 15 Try it Yourself This was just an example to demonstrate a simple function with different statements in C. The real power of a function is revealed in the next chapter, when we pass "parameters" to it. This allows the function to calculate the sum of any numbers, instead of being limited to the fixed values 5 and 10. In C programming, a function that does not take any arguments and does not return a value is called a void function. The syntax for defining a void function is as follows:

Syntax: return type function_name (parameter1, parameter2, ...) { // body of Statement; }

Example : void function_name () { // body of Statement; }

Here, the 'void' keyword is used as the return type to indicate that the function does not return any value. A void function can be called in the same way as other functions, by simply using the function name followed by empty parentheses function_name (). Here is an example of a void function that prints a message to the console. This program defines a void function called add() that simply prints the message "Total of a and b" to the console. The main function calls the add function, which executes the code inside the function and prints the message. Void functions are useful in C programming when you want to perform a specific task without returning any value, such as displaying a message, initializing a variable or updating a data structure.

Source Code //No Return Without Argument Function in C/* 1. Function Declaration 2. Function Definition 3. Function Calling */#include</FunctionDeclaration.h>

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
#include<math.h>
```

```
#include<string.h>
```

```
#include<ctype.h>
```

```
#include<limits.h>
```

```
#include<stropts.h>
```

```
#include<sys/types.h>
```

```
#include<sys/conf.h>
```

</div