

SNS COLLEGE OF TECHNOLOGY

Coimbatore – 35

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A++' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai


DEPARTMENT OF

INFORMATION TECHNOLOGY

23ITT101 – PROGRAMMING IN C & DS

UNIT II – DECISION STATEMENTS & FUNCTIONS

TOPIC – FUNCTIONS


A vertical column of decorative hexagonal icons on the left side of the slide. From top to bottom, the icons include: a lightbulb, a thumbs-up gesture, a large white double quote mark on a teal-to-blue gradient background, a smartphone, a magnifying glass, a gear, and a speech bubble. The hexagons are in various shades of blue and teal.

A function is a block of code which only runs **when it is called**. Functions are used to perform certain actions, and they are important for **reusing code**: Define the code once, and use it many times.

A decorative border of hexagonal icons is on the left side of the slide. The icons include a lightbulb, a thumbs-up, a network node, a smartphone, a magnifying glass, a gear, and a speech bubble. The hexagons are in various shades of blue and cyan.

1

DEFINITION OF FUNCTION

- 
- A cluster of decorative hexagonal icons in shades of blue and cyan. The icons include a thumbs-up, a lightbulb, a smartphone, and a magnifying glass.
- ✍ In c, we can divide a large program into the basic **building blocks** known as **function**.
 - ✍ The function contains the set of programming statements enclosed by **{ Curly braces }**.
 - ✍ A function can be called multiple times to provide reusability and modularity to the C program.
 - ✍ The function is also known as **procedure** or **subroutine** in other programming languages.

Advantages of Function

- ✍ By using functions, we can avoid rewriting same logic/code again and again in a program.
- ✍ We can call C functions any number of times in a program and from any place in a program.
- ✍ We can track a large C program easily when it is divided into multiple functions.
- ✍ **Reusability** is the main achievement of C functions.



Function Aspects

There are three aspects of a C function and they are,

Function Declaration

A function must be declared globally in a C program to tell the compiler about the function name, function parameters, and return type.

Function Definition

It contains the actual statements which are to be executed. It is the most important aspect to which the control comes when the function is called. Here, we must notice that only one value can be returned from the function.

Function Call

Function can be called from anywhere in the program. The parameter list must not differ in function calling and function declaration. We must pass the same number of functions as it is declared in the function declaration.

Types of Functions

There are two main types of functions in C programming. They are,

01

**STANDARD
LIBRARY
FUNCTIONS**

02

**USER – DEFINED
FUNCTIONS**

Standard Library Functions

- ✍ The standard library functions are also called as **Built - in functions** or **Pre - Defined functions** in C programming.
- ✍ There is a compiler package that already exists that contains these functions, each of which has a specific meaning and is included in the package.
- ✍ Built-in functions have the advantage of being **directly usable without being defined**, whereas user-defined functions must be declared and defined before being used.

Header files for Library Functions

- ✍ C has many libraries with pre-defined functions in the form of header files.
- ✍ To use these functions in our code, we need to include these header files.
- ✍ Header files contain **definitions of functions, macros, and data types** which we could use directly in our program by including the respective header file.

Example of Pre – defined functions along with its header files

<code>stdio.h</code>	<code>math.h</code>	<code>conio.h</code>	<code>string.h</code>
Standard input/output header file	Header file contains Mathematical functions	Console Input Output header file. Defines functions for formatting the output and getting the input in the console	Header file used to perform operations on the string
<code>printf()</code> , <code>scanf()</code> <code>puts()</code> , <code>gets()</code>	<code>sqrt()</code> , <code>pow()</code> , <code>log()</code> etc.,	<code>clrscr()</code> , <code>getch()</code>	<code>strcat()</code> , <code>strcmp()</code> , <code>strcpy()</code> , <code>strlen()</code>

Example Program



File Edit Search Run Compile Debug Project Options Window Help

STRING.C

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char a[100] = "SMS College";
char b[100] = " of Technology";
strcat(a,b);
printf("%s", a);
clrscr();
getch();
}
```

Header files for input/output functions, string function, console of input/output

strcat(a,b); String function of <string.h>

printf("%s", a); Output function of <stdio.h>

clrscr();
getch(); Console input/output of <conio.h>

SMS College of Technology_

21:1

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu


Advantages of Built – in functions

- ✍ The advantage of using inbuilt library functions in C are that we already have many of the functions we need pre-defined. This reduces the work of the programmer.
- ✍ It also makes code more efficient as the implementation of the pre-defined function may be more efficient than many of us can write.
- ✍ It makes the code more readable by separating the definition of functions from the actual code.

A cluster of hexagonal icons in various shades of blue and cyan is positioned at the top center. The icons include a smartphone, a magnifying glass, a lightbulb, and a thumbs-up gesture.

USER – DEFINED FUNCTIONS

A cluster of hexagonal icons in various shades of blue and cyan is positioned at the bottom center. The icons include a speech bubble, a gear, a network diagram, and a hexagon with a white border.

A vertical column of decorative hexagonal icons on the left side of the slide. From top to bottom, the icons include: a lightbulb, a thumbs-up gesture, a large white double quote mark, a smartphone, a magnifying glass, a gear, and a speech bubble. The hexagons are in various shades of blue and cyan.

C allows programmers to write their own functions, also known as **user-defined functions**. A user-defined function has three main components that are **function declarations**, **function definition** and **function call**. Further functions can be called by **call by value** or **call by reference**. Functions need to be written once and can be called as many times as required inside the program, which increases **reusability in code** and makes code more readable and easy to test, debug, and maintain the code.

A cluster of hexagonal icons in various shades of blue and cyan is positioned above the title. The icons include a smartphone, a magnifying glass, a lightbulb, and a thumbs-up gesture.

ELEMENTS OF USER – DEFINED FUNCTIONS

A cluster of hexagonal icons in various shades of blue and cyan is positioned below the title. The icons include a speech bubble, a gear, a network diagram, and a small hexagon.

There are 3 important elements in a user defined function



01

**FUNCTION
DECLARATION**

02

**FUNCTION
DEFINITION**

03

FUNCTION CALL

A vertical column of decorative hexagonal icons on the left side of the slide. From top to bottom, they include: a lightbulb, a thumbs-up, a network node, a smartphone, a magnifying glass, a gear, and a speech bubble. The icons are in various shades of blue and cyan.

1

DECLARING A FUNCTION

- ✍ A function declaration is **simply a prototype** of our function.
- ✍ Function declaration contains **function name, return type, and parameters** but does not contain the body of the function.
- ✍ Function declaration gives information to the compiler about the user-defined function that may be used in the later part of the code.

SYNTAX

`returnType functionName(type1 parameterName1,
type2 parameterName2, ...);`

Or

`returnType functionName(type1 , type2, ...);`



RETURN TYPE

The type of data returned from the function is called return type. A function may not return any output, in that case, we use void as the return type. In function declaration return type is mentioned before the name of the

function.



FUNCTION NAME

Function name is a unique name that can be used to identify our function in the program. A valid function name in C can contain letters, underscore, and digits; the first letter must not be a digit.



PARAMETER LIST

Parameters required by the function are also defined inside the declaration to tell the compiler number of arguments required by the function along with their data types.



SEMI COLON

Semicolon indicates the termination of a function declaration.




Example: `int getRectangleArea(int , int);`

A decorative border of hexagonal icons is on the left side of the slide. The icons include a lightbulb, a thumbs-up, a network node, a smartphone, a magnifying glass, a gear, and a speech bubble. The hexagons are in various shades of blue and cyan.

2

DEFINING A FUNCTION



Function definition contains the actual block of code that is executed once the function is called.

Components of Function Definition

01

Return Type

Type of Data returned from the function

02

Function name

Unique name of the function

03

Function Parameters

Parameters required by the function

04

Function body

It contains a collection of instructions that define what a function does

Syntax – Function Definition

```
returnType functionName(functionParameters...)  
{  
    // function body  
}
```

```
int getRectangleArea(int length=10 , int breadth=5)  
{  
    return length * breadth;  
}
```

A vertical column of decorative hexagonal icons on the left side of the slide. From top to bottom, the icons include: a lightbulb, a thumbs-up gesture, a network of nodes, a smartphone, a magnifying glass, a gear, and a speech bubble. The hexagons are in various shades of blue and cyan.

3

CALLING A FUNCTION

- ✍ To transfer the control to a user-defined function, we need to call the function.
- ✍ A function can be called using a **function name followed by round brackets**.
- ✍ We can pass arguments to function inside brackets, if any.

SYNTAX `functionName(functionArgument1, functionArgument2, ...);`

Creating a function call

- ✓ To call a function and calculate its output, we need to create a function call.
- ✓ Whenever a function is called, control of the program is transferred to called function, and the function body of the called function is executed.

```
int area = getReactangleArea(1, b);
```

Example Program



File Edit Search Run Compile Debug Project Options Window Help

```
STRING.C
#include<stdio.h>
#include<conio.h>

int getRect(int, int);

void main()
{
int l,b,area;
printf("Enter the value of Length and Breadth:");
scanf("%d", "%d", &l,&b);

area = getRect(l,b);

printf("Area = %d", area);
getch();
}

int getRect(int length, int breadth)
{
return length * breadth;
}
```

int getRect(int, int); FUNCTION DECLARATION

area = getRect(l,b); FUNCTION CALL

int getRect(int length, int breadth) FUNCTION DEFINITION

21:25

F1 Help Alt-F8 Next Msg Alt-F7 Prev Msg Alt-F9 Compile F9 Make F10 Menu



```
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
Enter the value of Length and Breadth:12
10
Area = 120
```

