Search this site...

HOME JAVA PHP Python Ruby Perl HTML CSS Javascript MySQL C++ UNIX MORE...

REFERENCES | FORUM | ABOUT | CONTACT



C Programming Tutorial

- C Home
- C Overview
- C Environment Setup
- C Program Structure
- C Basic Syntax
- C Data Types
- C Variables
- C Constants
- C Storage Classes
- C Operators
- C Decision Making
- C Loops
- C Functions
- C Scope Rules
- C Arrays

C - Variable Arguments

Advertisements

Previous Page Next Page

Sometime you may come across a situation when you want to have a function which can take variable number of arguments i.e. parameters, instead of predefined number of parameters. The C programming language provides a solution for this situation and you are allowed to define a function which can accept variable number of parameters based on your requirement. The following example shows the definition of such a function.

```
int func(int, ...)
{
    .
    .
    .
}
int main()
{
    func(1, 2, 3);
    func(1, 2, 3, 4);
}
```

It should be noted that function **func()** has last argument as ellipses i.e. three dotes (...) and the one just before the ellipses is always an **int** which will represent total number variable arguments passed. To use such functionality you need to make use of **stdarg.h** header file which provides functions and macros to implement the functionality of variable arguments and follow the following steps:

Define a function with last parameter as ellipses and the one just before the ellipses is always an **int** which will represent number of arguments.

Create a va_list type variable in the function definition. This type is defined in stdarg.h header file.

Use int parameter and va_start macro to initialize the va_list variable to an argument list. The

1 of 4 7/11/2013 7:46 PM

C - Pointers

C - Strings

C - Structures

C - Unions

C - Bit Fields

C - Typedef

C - Input & Output

C - File I/O

C - Preprocessors

C - Header Files

C - Type Casting

C - Error Handling

C - Recursion

C - Variable Arguments

C - Memory Management

C - Command Line Arguments

C Programming Resources

C - Quick Guide

C - Useful Resources

Selected Reading

Developer's Best Practices

Effective Resume Writing

Computer Glossary

Who is Who

macro va_start is defined in stdarg.h header file.

Use va_arg macro and va_list variable to access each item in argument list.

Use a macro va_end to clean up the memory assigned to va_list variable.

Now let us follow the above steps and write down a simple function which can take variable number of parameters and returns their average:

```
#include <stdio.h>
#include <stdarg.h>
double average (int num, ...)
   va list valist;
   double sum = 0.0;
   int i;
    /* initialize valist for num number of arguments */
   va start(valist, num);
    /* access all the arguments assigned to valist */
    for (i = 0; i < num; i++)</pre>
       sum += va arg(valist, int);
    /* clean memory reserved for valist */
    va end(valist);
    return sum/num;
int main()
  printf("Average of 2, 3, 4, 5 = fn", average(4, 2,3,4,5));
  printf("Average of 5, 10, 15 = %f n", average(3, 5,10,15));
```

When the above code is compiled and executed, it produces the following result. It should be noted that the function <code>average()</code> has been called twice and each time first argument represents the total number of variable arguments being passed. Only ellipses will be used to pass variable number of arguments.

```
Average of 2, 3, 4, 5 = 3.500000
Average of 5, 10, 15 = 10.000000
```

Previous Page Print Version PDF Version Next Page

Advertisements

2 of 4 7/11/2013 7:46 PM

Advertisements

ASP.NET | jQuery | AJAX | ANT | JSP | Servlets | log4j | iBATIS | Hibernate | JDBC | Struts | HTML5 | SQL | MySQL | C++ | UNIX

Copyright © 2013 by tutorialspoint. All Rights Reserved.

4 of 4 7/11/2013 7:46 PM