

# Command Line Arguments in C

## Overview

Command line argument is an important concept in C programming. It is mostly used when you need to control your program from outside. In C command line arguments are passed to the `main ()` method.

## Introduction to Command Line Arguments in C

Let's say we have a weather forecasting application developed in any language. When we run the program, it will display a graphical user interface (GUI) where you can enter the city name and hit the ENTER button to know about the current weather. But if you don't have that GUI, which means you can't click on any buttons that is where command-line arguments come into play, where we pass the parameters in the terminal box to do any actions.

### For example:

> weather "Delhi" and hit the ENTER key, this will show you the current weather.

In C programming, command line arguments are an important concept. Using command line parameters, we can perform any task. It is mostly used when you need to control your program from outside.

## What are Command-Line Arguments?

Command-line arguments are simple parameters that are given on the system's command line, and the values of these arguments are passed on to your program during program execution. When a program starts execution without user interaction, command-line arguments are used to pass values or files to it.

## What are Command-Line Arguments in C?

- When the main function of a program contains arguments, then these arguments are known as Command Line Arguments.

- The main function can be created with two methods: first with no parameters (void) and second with two parameters. The parameters are argc and argv, where argc is an integer and the argv is a list of command line arguments.
- argc denotes the number of arguments given, while argv[] is a pointer array pointing to each parameter passed to the program. If no argument is given, the value of argc will be 1.
- The value of argc should be non-negative.

## Syntax

- Main function without arguments:  
int main()
- \* Main function with arguments:  
int main(int argc, char\* argv[])

## Properties of Command Line Arguments in C:

- Command line arguments are passed to the main function as **argc** and **argv**.
- Command line arguments are used to control the program from the outside.
- **argv[argc]** is a Null pointer.
- The **name** of the program is stored in **argv[0]**, the first command-line parameter in **argv[1]**, and the last argument in **argv[n]**.
- Command-line arguments are useful when you want to control your program from outside rather than hard coding the values inside the code.
- To allow the usage of standard input and output so that we can utilize the shell to chain commands.
- To override defaults and have more direct control over the application. This is helpful in testing since it allows test scripts to run the application.

Example of Copy Program what we did in lab and why we used command line arguments

```
#include <stdio.h>

#include <stdlib.h>

#include <fcntl.h>

#include <errno.h>
```

#include <sys/types.h> - DATA types used in the system source code

#include <unistd.h> - to interface with API

#define BUF\_SIZE 8192

int main(int argc, char\* argv[])

int input\_fd, output\_fd; /\* Input and output file descriptors \*/

ssize\_t ret\_in, ret\_out; /\* Number of bytes returned by read() and write() \*/

char buffer[BUF\_SIZE]; /\* Character buffer \*/