**UNIT CONVERTOR**

SRI KIRUTHIKA CHANDRASEKAR

**TABLE CONTENTS:**

|  |  |  |
| --- | --- | --- |
| S.NO | TITLE | PAGE NUMBER |
| 1. | OBJECTIVE | 1 |
| 2. | INTRODUCTION | 1 |
| 3. | SOFTWARE USED | 1 |
| 4. | C PROGRAMMING | 1 |
| 5. | HEADER FILES IN C PROGRAMMING | 1 |
| 6. | INT MAIN() | 2 |
| 7. | VARIABLES AND DATATYPES | 3 |

|  |  |  |
| --- | --- | --- |
| 8. | PRINTF(), SCANF() | 4 |
| 9. | NEWLINE CHARACTER | 4 |
| 10. | ROLE OF SEMICOLON | 4 |
| 11. | IF-ELSE STATEMENT | 4 |
| 12. | C DYNAMIC MEMORY ALLOCATION | 5 |
| 13. | ARITHMETIC OPERATORS | 6 |
| 14. | C LIBRARY FUNCTIONS | 6 |
| 15. | PARAMETERS | 6 |
| 16. | FLOWCHART AND DESCRIPTION | 6 |
| 17. | SAMPLE OUTPUT | 7 |
| 18. | CONCLUSION | 8 |
| 19. | REFERENCES | 8 |

**OBJECTIVE:**

There are a few things as the following that we measure in our day-to-day activities, and they are:

* Speed
* Temperature
* Area
* Volume
* Mass

To measure these quantities,[units of measurements](https://byjus.com/physics/units-of-measurement/) are required. There are times when the units used for the measurement do not match the measurement preference and convenience as well as the standards prescribed for certain processes and applications. Converting such units to an extent that it can be understood directly and applied properly is important.

**INTRODUCTION:**

This mini project has been coded in C – Programming language. The operation involved in this project include Declaring variables, Choosing, allocating dynamic memory for the user input, and calculation for unit conversions.

**SOFTWARE USED:**

**C PROGRAMMING:**

C programming is a general-purpose, procedural, imperative computer programming language developed in 1972 by Dennis M. Ritchie at the Bell Telephone Laboratories to develop the UNIX operating system. C is the most widely used computer language.

**HEADER FILES IN C PROGRAMMING:**

In C language, header files contain the set of predefined standard library functions. The ***“#include”*** preprocessing directive is used to include the header files with ***“.h”*** extension in the program.

| **Sr.No.** | **Header Files & Description** |
| --- | --- |
| 1 | **stdio.h** - Input/Output functions |
| 2 | **conio.h** - Console Input/Output functions |
| 3 | **stdlib.h** - General utility functions |
| 4 | **math.h** - Mathematics functions |
| 5 | **string.h** - String functions |
| 6 | **ctype.h** - Character handling functions |
| 7 | **time.h** - Date and time functions |
| 8 | **float.h** - Limits of float types |
| 9 | **limits.h** - Size of basic types |
| 10 | **wctype.h** - Functions to determine the type contained in wide character data. |

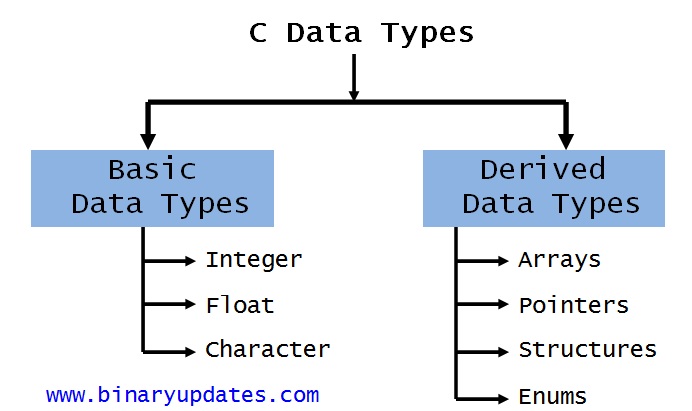
**INT MAIN() KEYWORD :**

An int is a keyword that references an integer data type. An int data type used with the main() function that indicates the function should return an integer value. When we use an int main() function, it is compulsory to write return 0; statement at the end of the main() function.

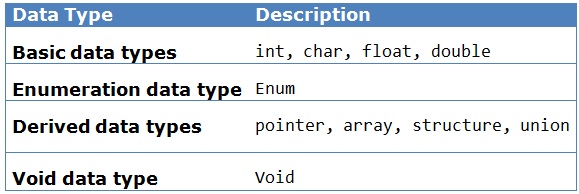
**VARIABLES AND DATA TYPES IN C PROGRAMMING:**

C is a strongly typed language. What this means it that, the type of variable cannot be changed. There are set of rules to be followed while declaring variables and data types in C Programming:

* The 1st letter should be alphabet.
* Variables can be combination of alphabets and digits.
* Underscore (\_) is the only special character allowed.
* Variables can be written in both Uppercase and Lowercase or combination of both.
* Variables are Case Sensitive.
* No Spaces allowed between Characters.
* Variable name should not make use to the C Reserved Keywords.
* Variable name should not start with a number.



**FIG-1 TYPES OF DATATYPES**



**FIG-1.1 DATATYPE DESCRIPTION**

**PRINTF() AND SCANF() IN C:**

The printf() and scanf() functions are used for *input and output* in C language. Both functions are inbuilt library functions, *defined in stdio.h* (header file).

The printf() function is used for output. It prints the given statement to the console. The scanf() function is used for input. It reads the input data from the console.

**The syntax of printf() and scanf() functions are given below:**

***printf("format string",argument\_list);***

***scanf("format string",argument\_list);***

**NEWLINE CHARACTER IN C: [\N]**

In programming languages, such as C, Java, and Perl, the newline character is represented as a '\n' which is an escape sequence.

**ROLE OF SEMICOLON IN C PROGRAMMING:**

Semicolons are end statements in C. The Semicolon tells that the current statement has been terminated and other statements following are new statements. Usage of Semicolon in C will remove ambiguity and confusion while looking at the code.

**C PROGRAMMING IF ELSE STATEMENT:**

The if-else statement in C is used to perform the operations based on some specific condition. The operations specified in if block are executed if and only if the given condition is true.

There are the following variants of if statement in C language:

* If statement
* If-else statement
* If else-if ladder
* Nested if

**IF-ELSE STATEMENT:**

The if-else statement is used to perform two operations for a single condition. The if-else statement is an extension to the if statement using which, we can perform two different operations, i.e., one is for the correctness of that condition, and the other is for the incorrectness of the condition. Here, we must notice that if and else block cannot be executed simultaneously. Using if-else statement is always preferable since it always invokes an otherwise case with every if condition.



**FIG- 2 FLOWCHART OF IF-ELSE STATEMENT.**

**C DYNAMIC MEMORY ALLOCATION:**

An array is a collection of a fixed number of values. Once the size of an array is declared, you cannot change it. Sometimes the size of the array you declared may be insufficient. To solve this issue, you can allocate memory manually during run-time. This is known as dynamic memory allocation in C programming. To allocate memory dynamically, library functions are malloc(), calloc(), realloc() and free() are used. These functions are defined in the <stdlib.h> header file.

The name ***"malloc***" stands for memory allocation. This function reserves a block of memory of the specified number of bytes. And, it returns a pointer of void which can be casted into pointers of any form.

**Syntax of malloc()**

ptr = (castType\*) malloc(size);

**Example**

ptr = (float\*) malloc(100 \* sizeof(float));

The above statement allocates 400 bytes of memory. It's because the size of float is 4 bytes. And, the pointer ptr holds the address of the first byte in the allocated memory.

**C - ARITHMETIC OPERATORS**

An arithmetic operator performs mathematical operations such as addition, subtraction, multiplication, division etc on numerical values (constants and variables).

**Operator - Meaning of Operator**

[ + ] - addition or unary plus

[ - ] - subtraction or unary minus

[ \* ] - multiplication

[ / ] - division

[ %] - remainder after division (modulo division)

**C LIBRARY FUNCTION VOID FREE(VOID \*PTR):**

The C library function void free(void \*ptr) deallocates the memory previously allocated by a call to calloc, malloc, or realloc.

**Declaration**

**Following is the declaration for free() function.**

void free(void \*ptr)

**PARAMETERS**

ptr − This is the pointer to a memory block previously allocated with malloc, calloc or realloc to be deallocated. If a null pointer is passed as argument, no action occurs.

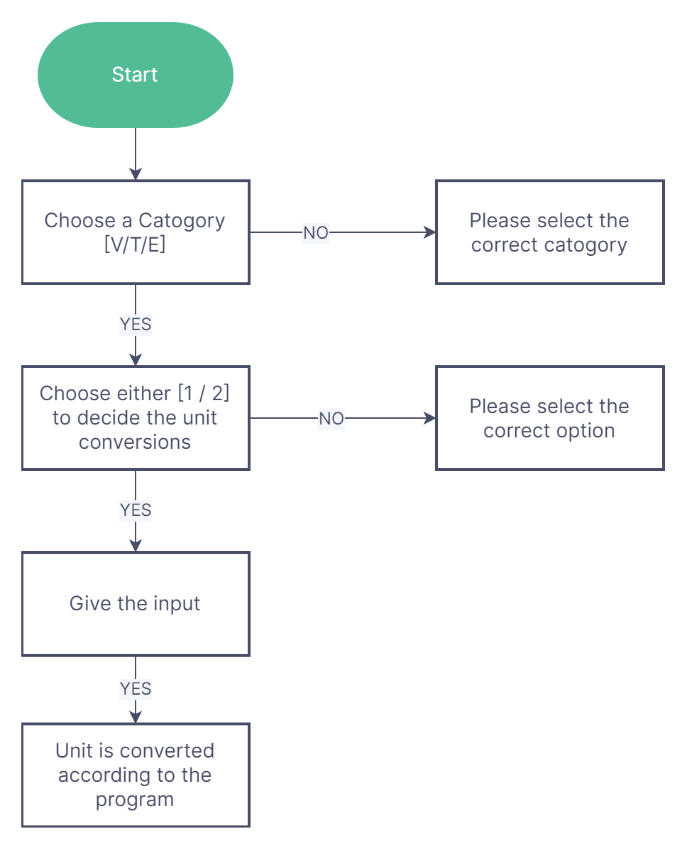
Return Value - This function does not return any value.

**FLOWCHART WITH DISCRIPTION:**

1. Program starts running.
2. Select the appropriate category as given in the Program [V/T/E].
3. After selecting, it will be asking you a question if it is either option 1 or 2.

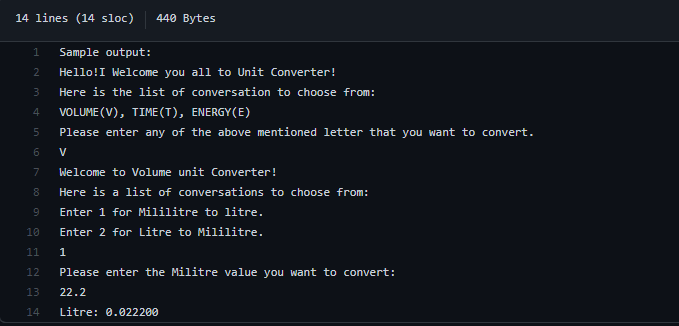
For Example, if the user selects “V” category and the chooses option 1 the program will convert the given input from Milliliter to liter if the user selects option 2 it is converted in a vice versa manner.

1. As the units are converted each value is allocated dynamic memory.
2. Program stops running.
3. When the program stops running the allocated memory is freed.



**FIG 3 – FLOWCHART**

**SAMPLE OUTPUT:**



**FIG – 4 SAMPLE OUTPUT**

**CONCLUSION:**

There are many conversion tools. They are found in the function libraries of applications such as spreadsheets databases, in calculators, and in macro packages and plugins for many other applications such as the mathematical, scientific and technical applications. There are many standalone applications that offer the thousands of the various units with conversions. For example, the free software movement offers a command line utility GNU units for Linux and Windows.

**REFERENCES:**

[1] Basics of C programming -- <https://www.programiz.com/c-programming>

[2] IF-ELSE Statement concepts -- <https://www.programiz.com/c-programming/c-if-else-statement>

[3] C programming pointers -- <https://www.programiz.com/c-programming/c-pointers>

[4] Memory allocation -- <https://www.programiz.com/c-programming/c-dynamic-memory-allocation>

[5] Standard Library functions -- <https://www.programiz.com/c-programming/library-function>

[6] Unit conversions -- <https://byjus.com/physics/unit-conversion/>

[7] Basic programs -- <https://beginnersbook.com/2015/02/simple-c-programs/>

[8] Practice -- <https://www.hackerearth.com/practice/basic-programming/input-output/basics-of-input-output/practice-problems/>

[9] Reference -- <http://www.wellho.net/resources/ex.php?item=c203/tall2.c>

[10] Memory allocation -- <https://www.javatpoint.com/dynamic-memory-allocation-in-c>