



C PROGRAMMING CHEATSHEET

About

Aptuts was founded by a software engineer with 12+ years of experience in IT industry.

Our online courses are built in partnership with technology experts and are relevant to the current education and industry needs.

www.apptuts.com

C Programming Course (86% Off)



[www.fb.com/Aptuts](https://www.facebook.com/Aptuts)



[Programming by Aptuts](#)

WHAT IS VARIABLE

A variable is an entity whose value may or may not change during the program execution

WHAT IS CONSTANT

A constant is an entity whose value can't be changed during program execution

WHAT IS KEYWORD

A keyword is the word reserved for C compiler and can only used to perform special task

VARIABLE CONSTRUCTION RULES

- A Variable name can consists of any combination of alphabets, digits and underscores
- The first character must either be alphabet or underscore. It should not start with the digit
- No commas and blanks are allowed

PRIMITIVE DATA TYPES

int, float, double and char

DATA TYPE SIZE

For 32-bit C compiler,

- int occupies 4 bytes of memory space
- float occupies 4 bytes of memory space
- double occupies 8 bytes of memory space
- char occupies 1 byte of memory space

WHAT IS MODIFIER

Modifier is used to re-define amount of memory space allocate to a variable

MODIFIERS IN C

short, long, signed and unsigned

- short can be used with int
- long can be used with int and double
- signed and unsigned can be used with int and char

COMMENTS IN C

Comments are used to explain or describe source code logic.

- Single line comment is represented by //
- Multi-line comments are enclosed with-in /* and */

VARIABLE DECLARATION

Syntax data-type variable_name;

Example int count;

ASSIGN VALUE TO A VARIABLE

Syntax variable_name = value;

Example count = 10;

BASIC C CODE

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

```
int main()
{
    // program body
    return 0;
}
```

TYPES OF INSTRUCTIONS

- Type declaration Instruction
- Arithmetic Instruction
- Control Instruction

TYPE DECLARATION

Type Declaration instruction is used to declare the type of variables used in C

Example float amount;

ARITHMETIC INSTRUCTION

Arithmetic instructions in C are used to perform arithmetic operations on variables and constants

Example total = 100 + balance;

ARITHMETIC OPERATORS

+ addition
- subtraction
* multiplication
/ division
% modulo division

CONTROL INSTRUCTION

Control instructions in C are used to alter the logical flow of program execution.

- if statement
- if-else statement
- conditional operator
- goto statement
- switch statement
- while loop
- do-while loop
- for loop
- break statement
- continue statement

IF STATEMENT

Syntax

```
if(condition)
{
    /* block of code */
}
```

If condition returns true, execute the block of code. Else, skip the block of code

IF-ELSE STATEMENT

Syntax

```
if(condition)
{
    /* 1st block of code */
} else {
    /* 2nd block of code */
}
```

If condition is true, execute 1st block of code and if condition is false, execute 2nd block of code

CONDITIONAL OPERATOR

Syntax condition ? result1 : result2;

If condition is true, evaluate result1 and return the value of result1. If condition is false, evaluate result2 and return the value of result2

SWITCH STATEMENT

Switch statement accepts expression and executes a particular case matching the expression value

Syntax

```

switch(expression) {
    case constant 1:
        //perform this;
    case constant 2:
        //perform this;
    .
    .
    .

    default:
        //perform this
}

```

WHILE LOOP

Syntax

```

while(condition)
{
    /*block of statement*/
}

```

Here, block of statement will be repeated until condition evaluates to false

DO-WHILE LOOP

Syntax

```

do
{
    /*block of statement*/
} while(condition);

```

Here, the block of statement after the keyword "do" is executed at least once. After the execution of block of statement for the first time, the condition in the "while" is evaluated.

Block of statement will be repeated until condition evaluates to false

BREAK STATEMENT

The break statement is used to break from any kind of loop and switch statement

syntax break;

Example

```

for(i=0;i<10;i++)
{
    /*block of statement*/
    if(i==5)
        break;
}

```

FOR LOOP

Syntax

```

for (initialization; condition; increment/decrement)
{
    /*block of statement*/
}

```

The for loop in C is executed as follows:

Step 1: The initial counter value is initialized. This initialization is done only once for the entire for loop

Step 2: Test condition is checked. Test condition can be any relational or logical expression. If the test condition is satisfied i.e. the condition evaluates to true then the block of statement inside the for loop is executed

Step 3: After the execution of the block of statement, increment/decrement of the counter is done

Step 4: After performing this, the test condition is again evaluated. The steps 2 and 3 are repeated till the test condition returns false

C Programming Course (86% Off)

GOTO STATEMENT

goto statement is used for unconditional jump from one part of the program to another part of the program

The goto statement consists of two parts: label and goto keyword.

CONTINUE STATEMENT

continue statement is used to force next iteration of the loop.

continue statement can be used with for loop, while loop and do-while loop

syntax continue;

Example

```
for(i=0;i<10;i++)
{
    /*block of statement*/
    if(i==5)
        continue;
}
```

FUNCTIONS

Functions are self contained block of statements that perform a particular task. C functions increases the readability of the program.

Example of a function that does not return a value and has no parameters

Syntax

```
void function_name()
{
    // Parameters are optional
    // block of statements
}
```

Example

```
void display()
{
    printf("This is a function");
}
```

This function can be called from main() using function name as follows:

```
{
    display();
}
```

FUNCTIONS (CONT.)

Example of a function that returns a value and with parameters

Syntax

```
data-type function_name(parameters)
{
    // Function with parameters
    // block of statements
}
```

Example

```
int add(int num1, int num2)
{
    return num1 + num2;
}
```

Function can be called using function name with parameters (if any)

Example

```
int main()
{
    int sum;
    sum = add(2, 3); //function call
    return 0;
}
```

ARRAYS

An array is a collective name given to a group of similar variables. Arrays can be 1-Dimensional, 2-Dimensional, 3-Dimensional and so on.

Syntax for 1-D array

```
data-type arrayName[count] = {value1,
value2, value3, ..., valueN};
```

Example of 1-D array

```
int arr[3]={10, 20, 30};
```

Here, arr[0] = 10, arr[1] = 20 and arr[2] = 30

Example of 2-D array

```
int A[3][3]={
    {11,12,13},
    {14,15,16},
    {17,18,19}
}
```

2-D array can be accessed using row and column index.

For example, A[0][0] will return value 11 similarly, A[1][1] will return value 15 i.e., value at row index 1 and column index 1.

Note that index in an array always starts with 0

STRUCTURES

Structure contains a number of data types grouped together.

Syntax

```
struct struct-name
{
    Element 1;
    Element 2;
    .
    .
    Element n;
} variable1, variable2, ..., variableN;
```

Example 1

```
struct animal
{
    int age;
    char gender;
};
//declaring structure variables
struct animal a1, a2;
```

Values can be accessed using a1.age, a1.gender, a2.age and a2.gender

Example 2 (declaring structure variables along with structure)

```
struct animal
{
    int age;
    char gender;
} a1, a2;
```

POINTERS

A pointer in C is a variable that represents the location (rather than the value) of a data item.

Syntax data-type *name;

Example int *p;

Here, I have declared a pointer variable having data-type int. This pointer *p will point to an address as shown below

```
int main()
{
    int a = 10;
    int *p;

    p = &a;
    //pointer p is now pointing to
    address of an int variable "a"

    //I can access value of a using *p
    printf("%d", *p);

    return 0;
}
```

STRINGS

A string in C is a series of characters in a group that occupy contiguous memory.

A group of characters (Alphabets, digits and special characters) is called as a string.

Example of a string

```
"This is a string"
```

A string in C should always be enclosed with in double quotes (")

Syntax for declaring string

```
char stringName[num_of_characters];
```

Example 1

```
char name[20]
```

The above statement declares an array named "name" (or string named "name") capable of holding 20 characters.

Example 2

```
char name[]="This is a string";
```

Here the size of an array is calculated automatically

STRINGS (CONT.)

Example 3

```
char *name = "This is another way to  
declare a string";
```

In order print a string, you can make use of printf statement with %s format specifier as shown below

```
printf(name);  
printf("%s", name);  
printf("%s",&name[0]);
```

FILE OPERATIONS

Opening a FILE

```
FILE *fp;  
fp = fopen("file.c","r");
```

The above statement would open file named "file.c" in "read" mode. It tells the computer that the file being opened would be used for reading purpose only.

FILE OPERATIONS (CONT.)

Reading from a FILE

```
ch = fgetc(fp);
```

To read the file's contents from the memory, there exists a function called fgetc().

fgetc() performs the following operations:

- Reads the characters from the current pointer position
- Advances the pointer position so that it now points to the next character
- Returns the character that is read, which we collected in the variable ch

Closing the FILE

```
fclose(fp);
```

The above statement will close the file

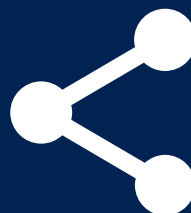


ABOUT

Aptuts was founded by a software engineer with 11+ years of experience in IT industry.

Our online courses are built in partnership with technology experts and are relevant to the current education and industry needs.

www.apptuts.com



FEEL FREE TO PRINT AND SHARE THIS CHEATSHEET ON C PROGRAMMING



WHY APTUTS?

- Take the online courses with the best personalized support
- Learn by doing. Our courses are hands-on centric
- We help you grow professionally
- 100% money back guarantee

C Programming Course (86% Off)



[www.fb.com/Aptuts](https://www.facebook.com/Aptuts)



[Programming by Aptuts](#)