

C Programming Cheat Sheet

Complete Edition | Calibri + Consolas | Colorized examples

Nikhil Learn Hub | nikhillearnhub.com

Code color legend

keyword string number comment #preprocess

1. BASIC SYNTAX

Part	Role
#include <stdio.h>	Enable printf / scanf
int main()	Program entry point
printf(...)	Print to stdout
return 0;	Success exit code

Syntax

```
#include <header>
int main(void) {
    /* statements */
    return 0;
}
```

Short example

```
#include <stdio.h>
int main(void) {
    printf("Hello World!\n");
    return 0;
}
```

Key points

- Statements end with a semicolon.
- Braces { } group a block.

2. VARIABLES & STORAGE

Kind	Meaning
Local	Inside a block; ends with block
Global	Outside functions
static local	Keeps value between calls
extern	Defined in another file
register	Register hint; no &addr

Syntax

```
type name;
type name = value;
```

Short example

```
int count = 10;
static int hits = 0;
extern int shared;
const int MAX = 100;
```

Key points

- Variable = named memory that stores a value.
- Names: letters, digits, _; cannot start with digit.

3. DATA TYPES

Category	Examples	Notes
Basic	char int float double void	Built-in
Derived	array pointer	From basic types
User-defined	struct union enum	Defined by you

Syntax

```
char c; int n; float f; double d;
unsigned int u; long int li;
```

Short example

```
char ch = 'A';
int age = 21;
float pi = 3.14f;
double big = 2.718281828;
```

Key points

- Modifiers: short, long, signed, unsigned.
- float ~6-7 digits; double ~15 digits precision.

4. IDENTIFIERS & KEYWORDS

Rule	OK?	Example
Start with letter or _	Yes	sum, _tmp
Contain digit later	Yes	val2
Start with digit	No	2val
Use a keyword	No	int, return

Syntax

```
identifier: [A-Za-z_][A-Za-z0-9_]*
```

Short example

```
int score_1 = 50;
// int = 5;    // invalid: keyword
```

Key points

- Keywords are reserved by the compiler.

5. FORMAT SPECIFIERS

Spec	Type
%c	char
%d / %i	signed int
%u	unsigned int
%f / %lf	float / double
%s	string
%p	pointer
%%	literal %

Syntax

```
printf("fmt", args...);
scanf("fmt", &vars...);
```

Short example

```
int n; char name[40];
scanf("%d %s", &n, name);
printf("n=%d name=%s\n", n, name);
```

5b. ESCAPE SEQUENCES

Escape	Meaning
\n	Newline
\t	Tab
\\	Backslash
\"	Double quote
\0	Null terminator

Syntax

```
char *s = "Line1\nLine2";
```

Short example

```
printf("Path: C:\\temp\\tOK\n");
```

6. OPERATORS

Type	Operators	Use
Arithmetic	+ - * / %	Math
Relational	< > <= >= == !=	Compare
Logical	&& !	Combine tests
Bitwise	& ^ ~ << >>	Bits
Assignment	= += -= *= /=	Store
Other	?: sizeof & * -> .	Misc

Syntax

```
a + b;  a == b;  a && b;  a & b;
x = cond ? t : f;
```

Short example

```
int a = 5, b = 2;
int sum = a + b;           // 7
int bit = a & b;           // 0
int m = (a > b) ? a : b;   // 5
```

7. CONDITIONALS

Form	When
if / else if / else	Multiple conditions
switch	One integer / enum value
?:	Inline choose expression

Syntax

```
if (cond) { }
else if (cond2) { }
else { }
switch (x) { case 1: break; default: break; }
```

Short example

```
int x = 10;
if (x > 0)
    printf("positive\n");
else
    printf("not positive\n");
int y = (x > 5) ? 1 : 0;
```

8. LOOPS & JUMPS

Construct	Notes
for	Entry-controlled; init;cond;update
while	Entry-controlled; may run 0 times
do-while	Exit-controlled; runs at least once
break	Leave loop / switch
continue	Next iteration
goto	Jump to label (avoid)

Syntax

```
for (init; cond; upd) { }
while (cond) { }
do { } while (cond);
```

Short example

```
for (int i = 1; i <= 5; i++) {
    if (i == 3) continue;
    printf("%d ", i); // 1 2 4 5
}
```

9. ARRAYS

Form	Example
1D	int a[5];
2D	int m[3][4];
Init	int a[3] = {1,2,3};
Length	n = sizeof(a)/sizeof(a[0]);

Syntax

```
type name[size];
type name[r][c];
```

Short example

```
int a[5] = {10, 20, 30, 40, 50};
a[2] = 100;
for (int i = 0; i < 5; i++)
    printf("%d ", a[i]);
```

Key points

- Index starts at 0. Size is fixed after declaration.

10. STRINGS

Function	Role
strlen	Length without '\0'
strcpy	Copy
strcat	Append
strcmp	Compare (0 = equal)

Syntax

```
char s[] = "text";
char buf[50];
```

Short example

```
#include <string.h>
char s[40] = "Hi";
strcat(s, " C");
printf("%s len=%zu\n", s, strlen(s));
```

11. POINTERS

Op	Meaning
&x	Address of x
*p	Value at p
p++	Next element
a[i]	Same as *(a+i)

Syntax

```
type *ptr = &var;
*ptr = value;
```

Short example

```
int x = 10;
int *p = &x;
*p = 20;           // x is now 20
printf("%d\n", x);
```

Key points

- Watch NULL, wild, and dangling pointers.

12. FUNCTIONS

Pattern	Example idea
Return + params	int sum(int a, int b);
No return	void show(int n);
Prototype	Declare before use

Syntax

```
return_type name(params);
return_type name(params) { body }
```

Short example

```
int sum(int a, int b) {
    return a + b;
}
int main(void) {
    printf("%d\n", sum(2, 3));
    return 0;
}
```

12b. DYNAMIC MEMORY

Call	Role
malloc(n)	n bytes, uninitialized
calloc(n, sz)	Zeroed block
realloc(p, n)	Resize
free(p)	Release

Syntax

```
#include <stdlib.h>
ptr = (type*)malloc(n * sizeof(type));
```

Short example

```
int *p = (int*)malloc(5 * sizeof(int));
if (!p) return 1;
p[0] = 42;
free(p);
p = NULL;
```

13. STRUCTURES

Access	Form
Variable	s.member
Pointer	ps->member

Syntax

```
struct Name {
    type member;
};
```

Short example

```
struct Point { int x; int y; };
struct Point p = {3, 4};
struct Point *q = &p;
printf("%d\n", q->x);
```

14. UNION & ENUM

Feature	union	enum
Idea	Shared memory	Named ints
Active	One member	N/A

Syntax

```
union U { int i; float f; };
enum Color { RED, GREEN, BLUE };
```

Short example

```
enum Color { RED, GREEN, BLUE };
enum Color c = GREEN; // 1
union Data { int i; float f; };
union Data d; d.i = 10;
```

15. FILE HANDLING

Mode	Meaning
r / w / a	Read / write / append text
r+ / w+ / a+	Update variants
...b	Binary mode

Syntax

```
FILE *fp = fopen("f.txt", "r");
fclose(fp);
```

Short example

```
FILE *fp = fopen("out.txt", "w");
if (!fp) return 1;
fprintf(fp, "score=%d\n", 95);
fclose(fp);
```

16. PREPROCESSOR

Directive	Use
#include	Insert header
#define	Macro / constant
#ifdef / #endif	Conditional compile
include guard	Stop double include

Syntax

```
#define NAME value
#ifndef H_H
#define H_H
#endif
```

Short example

```
#include <stdio.h>
#define PI 3.14159
#define MAX(a,b) ((a)>(b)?(a):(b))
double r = 2.0;
printf("%.2f\n", PI * r * r);
```

17. LIBRARY HEADERS

Header	Examples	Area
stdio.h	printf scanf fopen	I/O
stdlib.h	malloc free atoi	Memory
string.h	strlen strcpy	Strings
math.h	sqrt pow	Math (-lm)
ctype.h	isalpha toupper	Chars

Syntax

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

Short example

```
#include <math.h>
#include <stdio.h>
printf("%.2f\n", sqrt(9.0)); // link with -lm
```

18. INTERVIEW HOTSPOTS

Topic	Revise
Pointers + arrays	Decay, arithmetic, a[i]
Heap	malloc family + free + NULL
struct vs union	Padding vs shared store
Bitwise	set/clear; n&(n-1)

Syntax

```
n & (n - 1) // 0 if n is power of 2 (n>0)
```

Short example

```
int n = 8;
if (n > 0 && (n & (n - 1)) == 0)
    printf("power of two\n");
```

Educational summary for Nikhil Learn Hub. Topic map inspired by GeeksforGeeks C Cheatsheet; wording is original.