C LANGUAGE CHEATSHEET

from:
Çemberlitaş Anatolian High School
Software Club

Made by developers for developers with ❤

Variable Types

data type	signed	unsigned	format		byte
			signed	unsigned	
char	-128 +127	0 +255	%c	%c	1
int	-2,147,483,648 +2,147,483,647	0 +4,294,967,295	%d	%u	4
float	~±3.4e±38	–	%f	%f	4
double	~±1.7e±308	–	%lf	%lf	8
short	-32,768 +32,767	0 +65,535	%hd	%hu	2
long int	-9.22e+18 +9.22e+18	0 +1.84e+19	%ld	%lu	8
long double	~±1.1e±4932	–	%Lf	%Lf	16

Macros

```
#include <stdio.h>
#define LEVEL 2

#ifdef DEBUG
   printf("Debug mode on\n");
#endif

#ifndef MAX
#define MAX 100
#endif
```

```
#include "customLib"

#if LEVEL == 1
   printf("Easy mode\n");
#elif LEVEL == 2
   printf("Medium mode\n");
#else
   printf("Hard mode\n");
#endif
```

Functions

```
type functionName (type parameters) {
    instruction;
    return result;
}
int square (int number) {
   return (number * number);
}

void changeValue (int* number) {
   *number = 5;
}
```

follow us on Instagram: @cemberlitasyazilimkulubu
follow us on LinkedIn: linkedin.com/company/calyazilimkulubu

Conditions

```
if (condition) {
   instruction;
} else if (condition) {
   instruction;
} else {
   instruction;
}
```

```
switch (var) {
  case 1:
     instruction;
     continue;
  case 2:
     instruction;
     break;
  default:
     instruction;
     break;
}
```

```
max = (num1 > num2) ? num1 : num2;
```

Libraries

<stdio.h>	<assert.h>
<stdlib.h>	<limits.h>
<string.h>	<float.h>
<math.h>	<stddef.h>
<time.h>	<stdbool.h>
<ctype.h>	

File Operations

```
FILE *fp;
fp = fopen("file.txt", "w");
fputs("Hello, file!", fp);
-----------------------------------
fp = fopen("file.txt", "r");
char buffer[100];
fgets(buffer, 100, fp);
-----------------------------------
int ch;
while ((ch = fgetc(fp)) != EOF) {
   putchar(ch);
}
-----------------------------------
fp = fopen("file.txt", "a");
fprintf(fp, "Appending text\n");
fclose(fp);
```

Loops

```
for (initial; condition; update;) {
   instruction;
}
```

```
while (condition) {
   instruction;
}
```

```
do {
   instruction;
} while (condition);
```

Operators and Escape Sequences

Arithmetic	Logical	Comparison	Assignment	Bitwise
+ %	&&	>	+= &=	&
- ++	||	<	-= |=	|
/ --	!	>=	/= ~=	~
*		<=	*= ^=	^
		==	%= >>=	>>
		!=	= <<=	<<
		?:		

&	Address
*	Pointer Dereference
->	Access Thorugh Pointer
.	Access Structure Member
,	Seperate expression
sizeof	Find var's size in bytes
//	One line comment
/* */	Multi-line comment

New line	Carriage return	Horizontal tab	Null Character	Backslash
\n	\r	\t	\0	\\

Advanced Data Types

```
int array[5] = {10, 20, 30, 40, 50};
    ^        ^   ^           ^
  type      size Index 0   Index 4

char hello[6] = {'h', 'e', 'l', 'l', 'o', '\0'};

typedef struct {
   int year;
   char name[101];
} Car;

typedef enum {
   OK;
   ERR;
}Status;

union Car {
   int year;
   char name[101];
};
```

Memory Operations and Pointers

```
int variable = 5;
int* pVariable = &variable;

int *singleInt = (int*) malloc (sizeof(int) );
int *array = (int*) calloc (5 * sizeof(int));

int *resizedArray = (int*) malloc (3 * sizeof(int) );
resizedArray = (int*) realloc (resizedArray, 6 * sizeof(int) );

free(resizedArray);
```

String Operations

```
char str1[20] = "Hello";
char str2[20] = "World";

strlen(str1);
strcpy(str1, str2);
strncpy(str1, str2, 3);
strcat(str1, str2);
strncat(str1, str2, 2);
strcmp(str1, str2);
strncmp(str1, str2, 3);
strchr(str1, 'l');
strrchr(str1, 'l');
strstr(str1, "lo");

char *token = strtok(str1, " ");
token = strtok(NULL, " ");

char buffer[10];
memset(buffer, 0, sizeof(buffer));
memcpy(buffer, str1, 5);
memcmp(str1, str2, 5);
memmove(buffer + 2, buffer, 5);
```