Cheatography

C Programming Basics Cheat Sheet

by Scarlet_Cat via cheatography.com/210375/cs/45371/

How to Intialize an Variable

data_type **(var_name)**;

data_type **(var_name)** = initial_value;

Functions

Syntax:

```
return _type (func_ name) (datatype name1, datatype name2, ...) {
    .
    .
    return something;
}
```

```
int diff(int a, int b) {
 return a - b;
}
```

A function that takes two integers and performs subtraction

How to use Struct

```
struct (struc t_name) {
     (datatype) name1;
     (datatype) name2;
     .
 };
```

A struct is a type of pointer. Similar to making a class.

```
struct struct _name (name) = { val1, val2, ..}
;
```

Initializes a struct with the following values

```
p->x;
```

The arrow can be used to access members of a struct. This code accesses x from a pointer struct called p. Equivalent to (*p).x

<stdio.h> and <string.h> functions

strlen(string)	Counts the total char in a string	`char msg[] = " mes sag e";`
		`printf("%s", strlen(msg));`
		`// This code prints 7`

strcat(string1, string2)	Is used to join two strings together string1 adds the text of string2 to it	`char msg1[] = "My name is ";`
		`char msg2[] = "Vin";`
		`strcat(msg1, msg2);`
		`printf("%s", msg1);`
		`//This code prints "My name is Vin"`

strcmp()	Is used to tell if two strings are equal. Returns 0 if they are equal. Returns a non-zero value if they are not.	`char msg1[] = "xyz";`
		`char msg2[] = "xyz";`
		`int result = strcmp (msg1, msg2);`
		`printf("%d", result);`
		`//This code prints 0`

strchr()	Is used to find first occurrence of a char in a string	`char word = "wolves";`
		`void res = strchr (word, 's');`
		`printf("%s", res);`
		`//This code returns a pointer to s`

<stdio.h> and <string.h> functions (cont)		

strstr()	Is used to find first occurrence of a strin in a string	`char sent[] = "The world is cold";`
		`void* ptr = strstr (sent, "orl");`
		`if (sent != NULL) {`
		`printf("%s", ptr);`
		`}`
		`//This code prints if substring is found`

strcpy()	One string copies another	`char msg[] = "Jim";`
		`char* nmsg = malloc (st rle n(msg) * sizeof(char));`
		`strcpy(nmsg, msg);`
		`printf("%s\n", nmsg);`
		`//This code prints " Jim ", nmsg was changed`

DataTypes

char	refers to characters or strings	Ex. "car", 'c'
int	refers to an integer	Ex. 1, 2, 390
float	refers to a decimal, up to 6 digits	Ex. 1.909034
double	refers to a decimal, up to 15 digits	Ex. 1.9090341111
void	Empty	

Pointers

Description:	A variable that stores another variables address
`*`	The Dereference Operator:access the value stored at the address a pointer is pointing to
`&`	The Address Operator: It is used to get the memory address of a variable.
`int num = 6; int *ptr = &num;`	Return the address of num to ptr, and then dereference it. Essentially creating a pointer

Loops

for Loop	`for (int x = 0; x < 5; x++) { /body/ };`
while Loop	`while (x<5) { /body/ };`

How to use malloc()

Allocates space in memory of a specific block size. Returns a void pointer if successful. Remember to free the pointer when done.

`int nums = malloc (si zeo f(int) 10 );`	Creates an array of size 10

How to use realloc()

Re-allocated space of a given malloc() block space, will preserve data that's already there as long the new space is not smaller

`int* arr = malloc(5 * sizeof(int)); arr = reallo c(arr, 10 * sizeof (int) );`	The array that was created with malloc() is increased to be able to contain 5 more elements using realloc()

Arrays

Syntax:	*(datatype)* **(var_name)**[]
int arr[10];	An integer array with space for 10 integers
int nums[] = {1,2};	An integer array with elements declared
int arr[5] = {0};	An integer array of all zeroes
int arr3[5] = {1, 2};	An integer array, first two elements are set, others are 0
char word[] = "-Hello"	A char array which is basically a string

Format Specifiers

%c	Used for character data	char
%d	Used for signed integer data	int
%u	Used for unsigned integer data	unsigned int
%f or %.(num)f	Used for float or double, can insert a number before "f" for precision	float or double
%s	Used for string data	char (string)[] or char* (string)
%p	Used for printing the address of a pointer	void *(pointer)

Importing files

To import files use *#include ...* at the top of the file, these are .h files

Use < ... > if importing from standard c librarby

Use " ... " if importing your personal file

`#include <st dio.h>`	Standard Input Output library	printf(); scanf();
`#include <st rin g. h>`	A library with sting manipu-lation functions	strlen(); strcpy(); strcat(); memcpy(); memset();
`#include <st dli b. h>`	Standard Library	malloc(); realloc(); free(); rand();