STRUCTURE OF A PROGRAM IN C

Fabio Stroppa

Hello World!

First program: simply writes “Hello World” to your computer screen.

```
1: //  my first program in C
2: #include <stdio.h>
3:
4: int main()
5: {
6:    printf("Hello World!");
7: }
```

Hello World!

First program: simply writes “Hello World” to your computer screen.

```
1: //  my first program in C
2: #include <stdio.h>
3:
4: int main()
5: {
6:    printf("Hello World!");
7: }
```

This is where we will visualize code

Hello World!

First program: simply writes “Hello World” to your computer screen.

```
1: //  my first program in C
2: #include <stdio.h>
3:
4: int main()
5: {
6:    printf("Hello World!");
7: }
```

This is where we will visualize code

This is where we will visualize output

Hello World!

First program: simply writes “Hello World” to your computer screen.

```
1: //  my first program in C
2: #include <stdio.h>
3:
4: int main()
5: {
6:    printf("Hello World!");
7: }
```

Lines beginning with **//** are comments.

Hello World!

First program: simply writes “Hello World” to your computer screen.

```
1: //  my first program in C
2: #include <stdio.h>
3:
4: int main()
5: {
6:    printf("Hello World!");
7: }
```

Lines beginning with **#** are directives read and interpreted by the *preprocessor.*

Hello World!

First program: simply writes “Hello World” to your computer screen.

```
1: //  my first program in C
2: #include <stdio.h>
3:
4: int main()
5: {
6:    printf("Hello World!");
7: }
```

Lines beginning with **#** are directives read and interpreted by the *preprocessor.*

The directive **include** tells the preprocessor to access other functionalities, which are specified in the header **stdio**.

Hello World!

First program: simply writes “Hello World” to your computer screen.

```
1: //  my first program in C
2: #include <stdio.h>
3:
4: int main()
5: {
6:    printf("Hello World!");
7: }
```

Lines beginning with **#** are directives read and interpreted by the *preprocessor.*

The directive **include** tells the preprocessor to access other functionalities, which are specified in the header **stdio**.

stdio defines the standard input and output operations, such as writing the output of this program (Hello World) to the screen.

Hello World!

First program: simply writes “Hello World” to your computer screen.

```
1: //  my first program in C
2: #include <stdio.h>
3:
4: int main()
5: {
6:    printf("Hello World!");
7: }
```

Blank lines have no effect on a program but are good procedure for readability!

Hello World!

First program: simply writes “Hello World” to your computer screen.

```
1: //  my first program in C
2: #include <stdio.h>
3:
4: int main()
5: {
6:    printf("Hello World!");
7: }
```

Declaration of the **`main`** function.

Hello World!

First program: simply writes “Hello World” to your computer screen.

```
1: //  my first program in C
2: #include <stdio.h>
3:
4: int main()
5: {
6:    printf("Hello World!");
7: }
```

Declaration of the **`main`** function.

`main` is the function called when the program is executed.

Hello World!

First program: simply writes “Hello World” to your computer screen.

```
1: //  my first program in C
2: #include <stdio.h>
3:
4: int main()
5: {
6:    printf("Hello World!");
7: }
```

Declaration of the **`main`** function.

`main` is the function called when the program is executed.

All C programs start from the **`main`** function.

Hello World!

First program: simply writes “Hello World” to your computer screen.

```
1: //  my first program in C
2: #include <stdio.h>
3:
4: int main()
5: {
6:    printf("Hello World!");
7: }
```

Declaration of the **`main`** function.

`main` is the function called when the program is executed.

All C programs start from the **`main`** function.

For now, let’s ignore the meaning of **`int`** and of the parenthesis **`()`**.

Hello World!

First program: simply writes “Hello World” to your computer screen.

```
1: //  my first program in C
2: #include <stdio.h>
3:
4: int main()
5: {
6:    printf("Hello World!");
7: }
```

Body of the function, defined between braces:

- begin of the function defined by **{**
- end of the function defined by **}**

Hello World!

First program: simply writes “Hello World” to your computer screen.

```
1: //  my first program in C
2: #include <stdio.h>
3:
4: int main()
5: {
6:    printf("Hello World!");
7: }
```

C statement (i.e., instruction).

Hello World!

First program: simply writes “Hello World” to your computer screen.

```
1: //  my first program in C
2: #include <stdio.h>
3:
4: int main()
5: {
6:    printf("Hello World!");
7: }
```

C statement (i.e., instruction).

printf stands for print formatted, and it will allow to print messages on the default output device (i.e., the monitor).

Hello World!

First program: simply writes “Hello World” to your computer screen.

```
1: //  my first program in C
2: #include <stdio.h>
3:
4: int main()
5: {
6:    printf("Hello World!");
7: }
```

`printf` is defined in the header `stdio.h`

C statement (i.e., instruction).

`printf` stands for print formatted, and it will allow to print messages on the default output device (i.e., the monitor).

Hello World!

First program: simply writes “Hello World” to your computer screen.

```
1: //  my first program in C
2: #include <stdio.h>
3:
4: int main()
5: {
6:    printf("Hello World!");
7: }
```

C statement (i.e., instruction).

printf stands for print formatted, and it will allow to print messages on the default output device (i.e., the monitor).

“Hello World!” is the message to be printed.

Hello World!

First program: simply writes “Hello World” to your computer screen.

```
1: //  my first program in C
2: #include <stdio.h>
3:
4: int main()
5: {
6:    printf("Hello World!");
7: }
```

C statement (i.e., instruction).

`printf` stands for print formatted, and it will allow to print messages on the default output device (i.e., the monitor).

“Hello World!” is the message to be printed.

`;` defines the end of a statement.

Hello World!

First program: simply writes “Hello World” to your computer screen.

```
1: //  my first program in C
2: #include <stdio.h>
3:
4: int main()
5: {
6:    printf("Hello World!");
7: }
```

```
Hello World!
```

C statement (i.e., instruction).

`printf` stands for print formatted, and it will allow to print messages on the default output device (i.e., the monitor).

“Hello World!” is the message to be printed.

`;` defines the end of a statement.

Print “Hello World!” on screen.

Hello World!

First program: simply writes “Hello World” to your computer screen.

```
1: //  my first program in C
2: #include <stdio.h>
3:
4: int main()
5: {
6:    printf("Hello World!");
7: }
```

End of program.

```
Hello World!
```

Hello World!

First program: simply writes “Hello World” to your computer screen.

```
1: //  my first program in C
2: #include <stdio.h>
3:
4: int main() {printf("Hello World!"); }
```

start (a line of text) or position (a block of text) further from the margin than the main part of the text.

C does not have strict rules on **indentation** or on how to split instructions in different lines.

Proper text formatting helps readability.

```
Hello World!
```

Hello World!

First program: simply writes “Hello World” to your computer screen.

```
1: //  my first program in C
2: #include <stdio.h>
3:
4: int main() {printf("Hello World!"); }
```

start (a line of text) or position (a block of text) further from the margin than the main part of the text.

```
Hello World!
```

C++ does not have strict rules on indentatio… …w to split instruct…

Prop… …bility.

INDENTATION WILL BE EVALUATED IN YOUR PROJECTS!

Hello World!

First program: simply writes “Hello World” to your computer screen.

```c
//  my first program in C
#include <stdio.h>

int main()
{
   printf("Hello World!\n");
   printf("I am a C program");
}
```

```
Hello World!
I am a C program
```

Hello World!

First program: simply writes “Hello World” to your computer screen.

```
1: //  my first program in C
2: #include <stdio.h>
3:
4: int main()
5: {
6:    printf("Hello World!\n");
7:    printf("I am a C program");
8: }
```

The special character **'\n'** inserts a newline in the message.

```
Hello World!
I am a C program
```

Comments

```
//  line comments

/*  block comment */

/*
multiple
line
block
comment
*/
```

Line comments discard everything from where the pair of slash signs `//` are found up to the end of that same line.

Block comments , discard everything between the `/*` characters and the first appearance of the `*/` characters, with the possibility of including multiple lines.

Comments

```
1: //  my first program in C
2: #include <stdio.h>
3:
4: int main()
5: {
6:    printf("Hello World!\n");     //  prints Hello World!
7:    printf("I am a C program");   //  prints I am a C++ program
8: }
```

Writing comments is good practice!

```
Hello World!
I am a C program
```