



# C Bootcamp

## 24h Challenge

Staff WeThinkCode\_ [staff@wethinkcode.co.za](mailto:staff@wethinkcode.co.za)

*Summary: THE FOLLOWING TAKES PLACE BETWEEN 3.00 P.M. AND 4.00 P.M.*

# Contents

<b>I</b>	<b>Instructions</b>	<b>2</b>
<b>II</b>	<b>Foreword</b>	<b>4</b>
<b>III</b>	<b>ft_generic.c</b>	<b>5</b>

# Chapter I

## Instructions

- Only this page will serve as reference: do not trust rumors.
- Watch out! This document could potentially change up to an hour before submission.
- Make sure you have the appropriate permissions on your files and directories.
- You have to follow the submission procedures for every exercise.
- Your exercises will be checked and graded by your fellow classmates.
- On top of that, your exercises will be checked and graded by a program called Moulinette.
- Moulinette is very meticulous and strict in its evaluation of your work. It is entirely automated and there is no way to negotiate with it. So if you want to avoid bad surprises, be as thorough as possible.
- Moulinette is not very open-minded. It won't try and understand your code if it doesn't respect the Norm. Moulinette relies on a program called **Norminator** to check if your files respect the norm. TL;DR: it would be idiotic to submit a piece of work that doesn't pass **Norminator**'s check.
- These exercises are carefully laid out by order of difficulty - from easiest to hardest. We **will not** take into account a successfully completed harder exercise if an easier one is not perfectly functional.
- Using a forbidden function is considered cheating. Cheaters get -42, and this grade is non-negotiable.
- If `ft_putchar()` is an authorized function, we will compile your code with our `ft_putchar.c`.
- You'll only have to submit a `main()` function if we ask for a program.

- Moulinette compiles with these flags: -Wall -Wextra -Werror, and uses gcc.
- If your program doesn't compile, you'll get 0.
- You cannot leave any additional file in your directory than those specified in the subject.
- Got a question? Ask your peer on the right. Otherwise, try your peer on the left.
- Your reference guide is called `Google / man / the Internet / ....`
- Check out the "C Bootcamp" part of the forum on the intranet.
- Examine the examples thoroughly. They could very well call for details that are not explicitly mentioned in the subject...
- By Odin, by Thor ! Use your brain !!!

# Chapter II

## Foreword

24h Challenge initialisation...

Assets creation... OK

Background implementation... OK

Character's variable initialisation... OK

Extension Pack loading: Jersey Shore... OK

Extension Pack loading: Terrorist attach at CTU... OK

Extension Pack loading: White House... OK

Death rate adjustment to 84Decrease of Vic Soontobedead's life... OK

Terrorist meshed network creation...OK

Promotion of secondary characters... OK

Twilight operation launch... OK

3 compilation errors detected:

Minor error detected: Language variable set to FR-fr, should be EN-us. Language patch applied.

Minor error detected: Unnamed infiltrated traitor class variable. On the fly designation patch applied.

Minor error detected: 4th wall destroyed. No patch applied

3 compilation warnings detected:

Warning: Chloe O'Brian item unstable. Execution errors to be expected.

Warning: Vic Soontobedead's memory allocation overflow. Unauthorized access to data. Execution errors to be expected.

Warning: New York / White House flight plan data corrupted.


Program launched...

Internal clock update...

# Chapter III

## ft\_generic.c

The opening melody goes « Tu tu tu tu ; Tu tu tu tu ».

	Exercise 00
	ft_generic.c
Turn-in directory : <i>ex00/</i>	
Files to turn in : <b>ft_generic.c</b>	
Allowed functions : <b>write</b>	
Notes : n/a	

- Create a function `ft_generic()`. When called by a `main()`, it should display :

```
$>./a.out
Tu tu tu tu ; Tu tu tu tu
$>
```

- Here's how it should be prototyped :

```
void ft_generic(void);
```