Technische Universität München

Informatics I for MSE

MSE IN8011
Prof. Bjoern Menze, Giles Tetteh
Philipp Kaeß

WS 19/20
Homework 0

Due Date: 03.11.2019, 23:59

Release Date: 25.10.2019

Homework 0 : Hello World

Topics: Hello World, General Function Structure, Correct Upload

Introduction

This is only a dummy-homework and will **not** count towards your grade bonus. We recommend to complete this homework anyway as it will familiarize you with the submission system for the upload, the format of the provided code and how you receive your tutor's feedback.

A so-called *Hello World Program* has no functionality other than printing "`Hello World!`". This may sound completely useless, but is a good check if your coding environment is set correctly, e.g. your IDE and your compiler work.

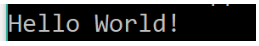
Specifications

When the function `say_hello()` is called by a main function, the following should happen:

- `Hello World!` is displayed

Example Output

Your terminal should look like that after an example run of your program:



```
Hello World!
```

Tool Box

The explanations below **may** help you to solve this problem.

PRINTING

You can print plain text with the function `printf()`. Simply put the text with double quotes in the brackets. To make a new line, use `\n`.

Example: `printf("This is the first, \nThis is the second line");`

Output:

```
This is the first,
This is the second line
```

COMPILE AND RUN

You can compile a C-File with the terminal by navigating to the respective directory with the *cd* command, and then calling ***gcc -Wall-Werror -o test main_hello.c***
Here, "test" is the name of the .exe-file the compiler should create, and "main_hello.c" is the name of the C-File you want to compile. After successful compilation you can run *test.exe* by *test* (Windows) or *./test* (Mac/Linux).

Tasks

We provide the template file ***hello_world.c*** on moodle. Complete the function `say_hello()`, so that the program fulfills the specifications above. Please note:

- You are only allowed to write code in the spot indicated in the function. Do not change anything else in the function. Write your name at the designated spot in the comment block at the beginning of your file.
- There is an additional C-File ***main_hello.c***. This file contains a main-function which you can use to test your function. To do this, store the two C-Files in the same directory on your computer, and compile and run **main_hello.c**. As in the beginning of this main file we stated `#include "hello_world.c`", the main file automatically includes all the code from your file when being compiled and therefore the main can run this function. Actually this is considered bad coding style, but as we didn't introduce headers yet, we will use this technique in the first few homeworks.

  Note: We use two different files for this homework to familiarize you with the template, even if it isn't necessary here. It will become important as we progress in the course, so we thought it would be best to introduce it now with an easy example.

- When you are finished, put the ***hello_world.c*** file **directly** in a zip-File named "HW0_Mustermann_Max" (with your name of course) and upload this file to Moodle. Do NOT upload your main-File and do NOT put any folders into the zip-File.

Threshold

To pass this homework, your program has to fulfill the following conditions:

- Your C-File must compile and run without any errors and with the flags `-Wall` and `-Werror` enabled
- It has to yield the correct output as specified above

Information on Plagiarism and Copyright

As already announced in the lecture and the tutorials, you are supposed to do this homework alone. This homework is an official part of the exam at the end of this semester, and therefore fraud or plagiarism will be pursued and punished.

You are encouraged to discuss **specific** problems with other students (like e.g. "I don't know how to print a string, can you help me?"), but you are forbidden to share ideas on the approach/structure of your functions, and of course to show other students your solution. This will result in a similar control-flow and structure of your submissions, which we will be able to detect even if you change variable names or loops etc.

We will compare your submissions with a code-optimized plagiarism scanner and if we find plagiarized code, you will fail this submission. If you are found guilty of plagiarism twice, you will lose the homework bonus, and in severe cases we will take further steps.

This homework is intellectual property of the TUM, namely the Chair of Image-Based Biomedical Computing. Publishing either this homework task itself or your solutions to it in any form (including on the MSE-Cloud) is strictly forbidden and can be pursued TUM-internally as well as legally.