

Exercise Sheet #2

Deadline: 29.04.2024, 12:00h

How to submit your solutions

- (a) In order to get the points for solving a problem you need to submit your fully **commented and compilable solution** in time. If you are unsure how to correctly comment your code, check out resources on the internet, e.g. <https://www.nickmccullum.com/how-to-write-cpp-comments/>.
- (b) For each problem, you are required to provide all relevant files, named in the format
`<YourName>_problem<number>_<additionalInfo>.xyz`.
Please do not use spaces in file names.
- (c) To hand in your solution to a sheet, send an e-mail to your tutor
mmoeller@itp.uni-frankfurt.de
konieczna@itp.uni-frankfurt.de
containing the bundled files from (b) to all problems named in the format
`<YourName>_sheet<number>.zip`.
- (d) Grades are given on a scale of 0 to 20 points. You are free to solve any of the given exercises, but 20 is still the maximal grade.
- (e) The **deadline for submission** is Monday of the week in which the sheet is discussed.

Problem 1 (*Gnuplot*) (5 points)

Create a gnuplot script following the instructions below:

- (a) Set up the terminal and output commands to create a PDF file with color as an output.

Hint: For output terminals see: <http://www.gnuplotting.org/output-terminals/>.

- (b) Create a plot script that ...

- ...sets the x -axis range to $(0, 10)$ and the labels of the x - and y -axes to x and $f(x)$, respectively.
- ...plots the two functions $\sin(x)$ and $\cos(x)$ in the same plot, using different line widths, line types and line colors. Give appropriate titles to each function.

Hint: For plotting functions see: <http://www.gnuplotting.org/plotting-functions/>.

- (c) Create the output file by calling `gnuplot <scriptname.gnu>`.

Problem 2 (Advanced: *Timer and Stopwatch*) (10 points)

Write a shell script that can be used as a timer or a stopwatch. Perform the following steps:

- (a) Ask the user to input `s` for the stopwatch or `t` for the timer. Check the users choice with an if-condition. If neither of the two options are chosen, display an error.
- (b) For the stopwatch (`s`), write code that counts up the elapsed seconds and displays them in the format `hh:mm:ss`.
- (c) For the timer (`t`), let the user input a time in seconds. Count down the time and display the remaining time to the user in the same format as above. When the time is over, display a message to the user.

Hint: The following cheat sheet, as well as the `man`-pages can help you with your solution: <https://github.com/LeCoupa/awesome-cheatsheets/blob/master/languages/bash.sh>

Hint: To run a program until it is aborted by the user using `Ctrl-C`, you can use `while true; do ... done`.

Hello World Program

Before starting the first problem in C++, it is advisable to test your development environment by running a *Hello World* program, as seen in the lecture. Create a file `helloworld.cpp` and copy the following code into it

```
#include <iostream>

int main() {
    std::cout << "Hello World!";
    return 0;
}
```

Compile and run the program using `g++ helloworld.cpp && ./a.out`.

Problem 3 (*Celsius to Fahrenheit Converter*) (5 points)

Converting temperatures in degrees Celsius ($^{\circ}\text{C}$) to degrees Fahrenheit ($^{\circ}\text{F}$) can be a bit cumbersome, since the conversion formula,

$$\text{Temperature in } ^{\circ}\text{F} = \left(\frac{9}{5} \cdot \text{Temperature in } ^{\circ}\text{C}\right) + 32,$$

involves **highly** complex mathematics! Ignoring the fact that there are a million simpler ways to do the conversion, we start writing a C++ program. Perform the following steps:

- Write a function `double celsius_to_fahrenheit(double celsius)` that takes a temperature in degrees Celsius and returns the corresponding temperature in degrees Fahrenheit.
- The final program should receive the temperature in degrees Celsius as a command line argument. With that in mind, implement a `main` function.
- Use your function from part (a) to convert the temperature given by the command line argument to a temperature in degrees Fahrenheit.

Problem 4 (*Odd or Even in C++*) (10 points)

Write a program that receives a number as input and checks whether it is even or odd:

- Use `cin`, `cout` to receive an integer number as input and output a message declaring it is odd/even.

- (b) Write an if-statement checking whether the number is odd or even.
Hint: You can use the modulo operator (%) to test if the number is even or odd.
- (c) **Optional:** Write the program as short as you can, using the least number of characters (or lines) possible.