General exercise information

During the semester, you can get a total of 0.25 bonus points in the final grade for solving the computational exercise sheets. You will get 0.25 bonus points for completing 80% of the exercises. The final grade is then the sum of the oral exam grade and the bonus grade (rounded and capped at 6.0). Participation in the bonus system is voluntary. It is possible to get a 6.0 without participating in the bonus system.

In order for exercise solutions to be counted towards the bonus, and for TA feedback to be guaranteed, they have to be handed in on time. New exercise sheets are uploaded on Thursdays and must be handed in before the start of the following lecture (the next Wednesday 09:45 Zurich time).

Handing in exercises is done via the ETH gitlab. The steps to be completed for each exercise are:

1. On the top level of your repository, create a folder `ex00` (with `00` replaced with the exercise sheet number) and put all the files part of the solution in there. You must include a scientific report containing your results, comments and figures in a single `.pdf` file. In case it's not obvious, please include instructions on how your solution can be run.
2. Once your solution is finished, make sure all files part of it are added to the git and push the final version of your solution to your gitlab repository.
3. Notify the TAs that your solution is complete and that you are handing it in. This can be done in two ways:
   - Either create an issue mentioning both TAs (@dgoekmen @engelerp)
   - Or send an email to both TAs stating that you are handing in your solution. The email addresses are `engelerp@phys.ethz.ch` and `dgoekmen@itp.phys.ethz.ch`. The subject of your email should start with `[csp22 handin]`.

Only once the TAs are notified will your solution be considered "handed in". In the issue text/email you should describe any potential issues or peculiarities you may have observed with your code, such that you can receive targeted feedback. The officially supported programming languages for this course are `C++` and `Python`. The teaching assistants will provide feedback and help with the conceptual design and logic of your code if it is written in either of these languages. Keep in mind, however, that the teaching assistants are not meant to make detailed corrections of your code or assist with debugging/setup on each specific platform.

Handing in as a group

Exercises can be handed in as a group of up to 3 students. In this case, the contributions of each member of the group must be clearly identified in the report, and all members of the group must be named in the issue text/email that announces the hand in. Each member of the group

must still hand in individually (although the solutions are assumed to be identical). Keep in mind that plagiarism is a serious offence and will not be tolerated.

The next section shows how necessary core tools can be installed, and the section thereafter is intended as a walkthrough on getting git and gitlab set up.

If at any point you have issues handing in your exercise solutions, please contact the TAs.

Installing core tools

There are a few tools that you will need in this course. In order to hand in exercises you will need the version control tool `git`. Depending on your choice of programming language you will need `python` along with `pip` and a few packages, and/or a `C++` compiler. Apart from those tools, you will also need a text editor to develop your programs. A few examples are TextMate, Kate, Gedit, Geany, Atom and vi.

The following shows installation pointers for various operating systems. Pick what you need.

MacOS:
The tools are most easily installed using a package manager, e.g. `homebrew`.

- Install `homebrew` using the instructions provided on https://brew.sh
- Install `git`: `brew install git` and verify that the installation was successful by issuing `git --version`
- Install `python`: `brew install python@3.8`
- Install `python` packages using `pip`: `python3.8 -m pip install numpy matplotlib`
- Install `gcc`: `brew install gcc`

Linux:
Use your distribution's package manager. On Ubuntu:

- Update package lists: `sudo apt-get update`
- Install `git`: `sudo apt-get install git` and verify that the installation was successful by issuing `git --version`
- Install `python`: `sudo apt-get install python3.8`
- Install `python` packages using `pip`: `python3.8 -m pip install numpy matplotlib`
- Install `gcc` and `Make`: `sudo apt-get install build-essential`

Windows:
We do not officially support Windows. One possibility is for you to use the Windows Subsystem Linux (WSL) to get a Linux shell. Then you can follow the Linux specific instructions.

If you do want to use Windows (without our support), here are a few pointers: There is a `git` implementation available for Windows, see here: https://git-scm.com/download/win. You can

verify your installation by opening a command prompt and issuing `git --version`. A recent version of python can be installed from the Microsoft Store. In order to develop and compile `C++` code, the Community Edition of Visual Studio could be used.

Git setup

This section is a very brief introduction to how we can set up `git` such that we can use it to hand in exercises. You will first set up a remote repository, which you then clone to a local repository on your machine. Finally you will test handing in exercises.

It is assumed that you are familiar with using the shell. If this is not the case, you may first want to have a look at a shell tutorial.

It should be noted that there's much more to version control with `git` than what's presented here. If you have never used `git` before, it is strongly recommended that you read up on it (e.g. you could start by having a look at a git tutorial).

Creating a remote git repository

Now you will create a remote git repository to which your solutions will be uploaded.

1. In your webbrowser of choice, navigate to https://gitlab.ethz.ch
2. Login with your `nethz` credentials
3. On the top right, click the `New Project` button
4. Click `Create blank project`
5. Fill in the `Project name` sensibly (e.g. csp22ex), and be sure to leave the `Initialize repository with a README` checkbox at the bottom checked. Click `Create project`.
6. In the top right, press the blue `Clone` button and copy the `https` link. You will need it later when cloning the repository.

Now that your repository exists, you have to give the TAs access rights such that they can inspect your handins.

1. With your repository still open, on the top left click on `Project information`, then `Members`.
2. Give `Maintainer` rights to `@dgoekmen` and `@engelerp`

One more thing you may want to set up are SSH keys, such that you don't have to log into gitlab each time you pull/push via SSH. This is not required, but instructions can be found here: https://gitlab.ethz.ch/help/ssh/index.md. If you do so, you may want to clone via SSH rather than HTTPS.

Cloning the remote repository to your local machine

Now that we have a remote repo, we need to clone it to our local machine.

1. Make sure you still have the `https` link from the previous instruction copied
2. In your terminal/console/command prompt, navigate to a directory where you would like your repository to sit
3. If this is your first time running `git`, you will need to do a bit of setup. Issue:
   - `git config --global user.name "Your Name"`
   - `git config --global user.email userid@ethz.ch`

   possibly with `Your Name` and `userid` replaced by useful strings
4. Clone the repository by issuing `git clone https://gitlab.ethz.ch/engelerp/csp22.git` with the link replaced with your own repo's `https` link. You will need to enter your `nethz` credentials.
5. In your current directory, there is now a new folder that contains the repo. In my example it's called `csp22ex`.

Test handing in exercises

In order to verify that you can hand in exercise solutions, we'll now hand in something for exercise 0.

1. Within your repo's folder (in my example `csp22ex`) create a new folder named `ex00`
2. In this folder we want to place all files associated with exercise 0. For now, just add a file `testing.txt` that contains some text.
3. In your terminal/console/command prompt navigate into the repo folder (in my example `csp22ex`)
4. Add the new exercise folder and all files within by issuing `git add ex00`
5. Commit your changes by issuing `git commit -am "Exercise 0 done!"`
6. Push your changes to the remote repository by issuing `git push`
7. Navigate to your project on https://gitlab.ethz.ch. You should see the new folder and file.

In order to notify the TAs that you have handed in your solution, you must create an *issue* in your repository.

1. Open an issue by clicking the `Issue` tab in the left sidebar, then `New issue`
2. Add a title, and in the description write down relevant information and mention both TAs (`@dgoekmen` `@engelerp`). Click `Create issue`. The TAs will then be notified of your issue.

If everything worked, you have now successfully handed in exercise 0.