# Übungen zu Algorithmen und Programmentwicklung für die Biologische Chemie

Michael T. Wolfinger

Research group Bioinformatics and Computational Biology
Department of Theoretical Chemistry
Währingerstrasse 17, 1090 Vienna, Austria

Summer 2021

# Contact

**Michael T. Wolfinger**

TBI—Theoretical Chemistry
Währingerstrasse 17
1090 Vienna

| | |
|---|---|
| mail: | `mtw@tbi.univie.ac.at` |
| Github: | mtw |
| web: | `https://www.tbi.univie.ac.at/~mtw` |

# Tutorial

## Übungen zu Algorithmen und Programmentwicklung für die Biologische Chemie (APBC)

(Lab class 'Algorithms and program development for biological chemistry')

| | |
|---|---|
| **Aim:** | improve programming skills<br>(while implementing interesting stuff) |
| **Credit:** | 4.0 ECTS credits |
| **Time:** | Wednesdays 10–12/13<br>meetings every (second) week |
| **Your job:** | • solve assignments<br>• use git/Github<br>• discuss solutions during course<br>• final presentation |

# Tentative Dates and Topics

(subject to change depending on COVID-19 situation)

| | |
|---|---|
| Mar-03-2021 | Github submission and warm up assignment: Word Count |
| Mar-10-2021 | Optimization |
| Mar-24-2021 | Dynamic programming |
| Apr-14-2021 | Random Sequences |
| Apr-30-2021 | Topic to be defined later |
| May-12-2021 | Team Project (planning meeting) |
| May-26-2021 | Team Project II (coordination meeting) |
| Jun-23-2021 | Final presentations |

Remember: besides specific algorithms (and instead of learning "programming" itself), you'll learn practical programming skills (reading/writing files, command line parsing, using version control etc.) and get programming experience.

# Scripting vs. Compiled Language

1. Scripting:
   - + Easier to implement small programs
   - + Implicict memory handling
   - + Fast application development (no compilation)
   - + Built-in powerful data structures (e.g. hash in Perl, dictionaries)
   - ± Dynamic typing (i.e. variables can hold any type of data)
   - – Speed
   - – Less enforcement of how to code (harder to read)
2. Compiled:
   - + Speed (precompiled as machine code)
   - + Type safe (declaration of variables)
   - + Static typing (efficient programs)
   - + Memory handling (efficient programs)
   - – Typically more code for simple tasks
   - – More responsibility for developers (memory leaks, null pointers, etc.)
   - – Write-Compile-Test-Cycle takes time

# Which language to choose?

Solutions are accpeted in any* scripting or compiled programming language.

**If you are open minded ( or simply undecided ):**

- if this is your first language: try Python
- if you already know Python: try Perl
- if you already know scripting languages like Perl or Python well: try a compiled language like C++ and vice versa. Take the opportunity to learn something new!
- if you are fluent in several languages:
  choose *the right language* for every task
  Feel free to change between languages.

# Submission System

**We will use git (`http://git-scm.com/`) and its online project hosting platform github (`www.github.com`).**

- easy to use revision/version control system
- git is free software; github is free of charge for open source projects
- ideal for collaborative software development
- available for all operation systems
- supports automatic testing of your programs (continuous integration test)

# Our Github repository

- Go to

  `https://github.com/TBIAPBC/APBC2021`
- This is our repository APBC in our Github organisation TBIAPBC.
- Essentially, this is a collection of files—together with their history.
- Scroll down to have a look at the README.md text. Here, one finds general information about the class and some about how to go on.

# Git/Github Warm Up

- Go to directory A0.
  You can click on directory A0 or surf to

  `https://github.com/TBIAPBC/APBC2021/tree/master/A0`

- This is assignment '0'; we use it as a first submission test here
- The assignment description is found in README.md
- Please follow the command line way, even if many IDEs support git
- Go through the steps yourself (at home or start now)

# The first assignment—Counting Words

- Go to directory A1.
- Follow the instructions: write the program and submit it by beginning/mid of next week.

# Forget Moodle—Github rules

- Use Github to post questions, problems, etc. In case, mention a particular user (e.g. @mtw), which will trigger notification of this person.
- Important dates and information are posted in the main README.md of our Github repo.
- Assignments are posted in respective subdirs
- Issues are there to discuss general interest questions.
- PR comments are there to discuss single submissions.
- Github is a collaborative place: **Involve yourself in discussions and code review.**

**Happy Hacking!**

**See you next week!**