Student first & last name:

______________________________

Student matriculation num:

# Welcome to the examination for *Introduction to Programming and Computational Thinking for Lawyers* on 18 December 2023, from 09:00 to 10:20

**Please read carefully!**

## 1 General information about the examination

- Answer the questions on this paper using the next **5** pages.
- Write your name and student ID on *every* paper.
- Start of examination: **09:00 AM**, Dec 18, 2023
- Examination time: **70 minutes**
- Maximum number of points achievable in the examination: **70 points**
- The distribution of questions/points is as follows:
  - 30 points from single choice question / 6 questions (5 points each)
  - 40 points from text questions / 4 questions (10 points each)
- Unless otherwise noted, the notation and the assumptions made in the lectures and assignments apply to the questions.

## 2 Resources

- This is an open book examination, i.e., you can access external learning materials, but you do not have to; in any case, you must adhere strictly to the time constraints.
- You cannot——under any circumstance——communicate with other people or use any form of Generative AI (e.g., GitHub Copilot and ChatGPT) during this examination.

## 3 Examination questions

### 3.1 Single-choice questions (6 questions, total 30 points)

- For single-choice questions there are **5** statements, of which only **one** is **applicable.**
- For all questions**,** always **select** ONE **correct** answer.
- **Evaluation**: Correctly answered questions result in **5** points, no/wrong answer 0 points.

### 3.2 Text questions with open-answers (4 questions, total 40 points)

- Keep answers short.
- If the question asks you to write code, make sure that the indentation level is clearly recognizable, because indent has a syntactical meaning in Python.

## 4 Confirmation

By starting and submitting your exam, you are confirming the following:

*“I hereby confirm that I will complete this official assessment myself and will not receive any inadmissible support.”*

University of Zurich$^{\text{UZH}}$

Student first & last name:

________________________________________

Student matriculation num:

## 1 Single-choice questions

Which of the following is *not* a programming language, rather a markup language?

- Python
- Java
- C++
- [✓ HTML]
- Swift

What is the *main* purpose of modules/packages in programming?

- To make the code shorter
- To make the code more readable
- To make debugging easier
- [✓ To reuse code]
- To make the program run faster

What is *false* about Python's `for` loop?

- It is used to iterate over a sequence.
- It can iterate over a list or dictionary.
- [✓ It can be used with an else clause, executed when the loop has finished.]
- It can be used with a `break` statement to stop the loop before it has looped through all the items.
- It can be used to iterate over a string.

What is *true* about dictionaries in Python?

- [✓ a dictionary is a mutable data structure.]
- a dictionary is an immutable data structure.
- a dictionary is a data structure to store sequences.
- a dictionary is a data structure to store characters.
- a dictionary is a data structure to store boolean values.

Which of the following is *not* a built-in data structure in Python?

- List
- Dictionary
- Tuple
- [✓ Tree]
- All the others are built-in data structures

How does a Python dictionary handle duplicated keys?

- By storing both values in a list.
- By ignoring the new key-value pair.
- By creating a linked list at that key.
- [✓ By overwriting the existing key-value pair.]
- By raising a KeyError exception.

University of Zurich^UZH

Student first & last name:

______________________________

Student matriculation num:

## 2 Text questions with open-answers

### 2.1 Programming Environment

Using the terminal, you can (1) start Python as in `python3` then write commands line by line or (2) use Python as in `python3 my_program.py`.

a. Is using Thonny more like the first or the second case? Why?

b. Describe two scenarios: one in which (1) is a better choice, and one in which (2) is a better choice.

Write your answer here:

a. Using Thonny is more similar to the second case, because Thonny calls python3 on the file currently saved in the editor.

b. (1) is a better choice when testing commands and prototyping and when interaction with data is necessary. (2) is a better choice with more stable code that also includes definition of functions.

Student first & last name:

Student matriculation num:

## 2.2 Lists

Write a function multiply_by_largest. It takes a parameter named numbers, which contains a list of numbers (excluding the number `0`).[1]

The function finds the largest number in the list. Then the function multiplies each number in the list by such largest value. Finally, the function prints the content of the list to the terminal.

ATTENTION: The function does not create a new list! Instead, it must modify the numbers *directly within the list* it receives as parameter.

**Example:** multiply_by_max([23,11,3]) will print "[529,253,69]".

Write your answer here:

```
def multiply_by_largest(numbers):
  max_num = max(numbers)
  for idx, value in enumerate(numbers):
    numbers[idx] = value*max_num
  print(numbers)
```

[1] Do not check that the list contains numbers and excludes the 0 value–you can assume it is correct.

**Student first & last name:**

______________________________________________

**Student matriculation num:**

## 2.3 Dictionaries

Write a function most_occurrences that takes a dictionary as a parameter. This dictionary contains law articles as keys and how many times these law articles are mentioned in a legal text as values.[2] The function should return a list of all the articles that appear the most.

ATTENTION: You are not allowed to use any built-in Python functions or methods to find the maximum value. Instead, you can use loops and conditional statementdds.

**Example:**

most_occurrences({'Art. 21':5, 'Art. 19':2, 'Art. 16a':5}) will return ['Art. 21','Art. 16a'].

Write your answer here:

```
def most_occurrences(dic):
    max_occ = max(dic.values())
    l = []
    for k,v in dic.items():
        if v == max_occ:
            l.append(k)
    return l
```

[2] Do not check that the dictionary is in this format–you can assume it is correct.

## 2.4 Writing to files

Write a function `strings_to_file`, which takes two parameters:[3] a list of strings `list_strings` and a string that represents a the full name of an existing file `fn `.

The function writes the strings in the list into the file, one per line, each preceded by the current line number (the first line is number `0`). Also, the function returns the number of lines written to the file.

**Example:** If we call strings_to_file(['computational','thinking']), then the file will contain 0 computational in the first line and 1 thinking in the second line, and the function will return 2.

Write your answer here:

```
def strings_to_file(list_strings,fn):
    f = open(fn,'w')
    counter = 0
    for string in list_strings:
        f.write(f"{counter} {string}\n")
        counter = counter + 1
    f.close()
    return counter
```

[3] Do not check the format of the parameters–you can assume they are correct.