Name of Course: **Introduction to Programming in Python**

Lecturer: **Petra Kralj Novak**

No. of Credits and no. of ECTS credits: **US Credits: 3; ECTS Credits: 6**

Semester or Time Period of the course: **Fall 2022, starting on September 26, 2022**

The status of the course: **Mandatory for 1st year students of QSS with no pre-requisites**. The course is one of the introductory courses of the module Programming and Data Analysis. Later courses of the same module heavily build on the knowledge gained here, namely Fundamentals of Data Analysis, Algorithms and Data Structures, Machine Learning, Data Management and Data Science Project.

Classes will be a combination of lectures and hands-on sessions. Office hours will be available for clarifications.

The aim of the course

The aim of this course is to provide an overview of key programming concepts using the Python programming language. Python is a general-purpose programming language used for a wide variety of tasks, and the language of choice for Data Science. In this course, students will be introduced to programming by writing their first "Hello World!" program and become proficient Python programmers, able to tackle a variety of programming tasks.

Learning outcomes

After completing this course, the students will know the basics of computer programming. They will be familiar with the basic syntax and use of the Python programming language, including writing and executing scripts to automate common tasks using Jupyter notebook and the PyCharm IDE environment. Students will first get familiar with variables, expressions, functions, logical expressions, conditions, and control flow. They will learn about simple data types with an emphasis on strings. Then they will be introduced to complex data types, including lists, tuples, dictionaries, and sets. After learning the basics, the course continues with functions, file input and output, and object-oriented programming. Students taking this course should be able to write complex scripts for their research projects using the combination of codes they covered at class.

Detailed content

Jupyter notebooks for each class will be available on Moodle. All notebooks will consist of explanatory sections and associated smaller hands-on assignments designed to better explain the introduced concepts and code. After each main topic, students will be given a complex assignment for the hands-on session, with available advice and assistance from the instructor and TA. Homework will be assigned and graded weekly. The solution of the homework will be discussed in class.

Weekly breakdown:

Week 1: Basics
Running Python scripts, print, constants and variables, mathematical expressions, user input, data types, introduction to Jupyter notebooks and PyCharm
Week 2: Conditional Expressions
Conditional executions; alternative executions; chained and nested conditionals
Week 3: Lists and loops
Complex data types list and tuple and for loops to iterate over them
Week 4: Functions
Defining custom functions, function parameters, returning values, packing functions into modules
Week 5: Methods and name spaces
Built-in methods of lists and strings
Week 6 and 7: Dictionaries and Sets
Complex data types based on key-value pairs for easy access to stored data
Week 8: File input and output
Reading and writing files, file types, parsing text files
Week 9 & 10: Object-oriented programming
Week 11: Recursion
Week 12: Recap and Final Evaluation

Recommended reading
Open book: Non-Programmer's Tutorial for Python

Software
Jupyter Notebook in the environment of Anaconda, freely downloadable here (download the version, which contains Python 3.8): https://www.anaconda.com/download/

PyCharm Community edition is an integrated development environment (IDE) for programming: https://www.jetbrains.com/pycharm/download/

Assignments
Homework will be assigned weekly with a deadline in the same week. Two homework assignments will be larger and more time will be available to submit them.

Assessment
The final grade is composed as follows:

- 10% small homework assignments
- 20% first large homework
- 20% second large homework
- 50% final evaluation: large programming exercise in class

Retake
The retake is in the same form as the final exam and accounts for 100% of the final grade. Refer to the CEU Undergraduate studies handbook for grading.