UNIVERSITY OF NEW YORK IN PRAGUE

Course: **ITM351 Introduction to Python** (6 ECTS/3 US credits)
Semester: Fall 2023
Prerequisites: Algorithms
Instructor:

1. Course Purpose

The course introduces students to programming via programming language Python. Since many programming techniques are transferable between various programming languages, this course also focuses on algorithms and data structures design.

Python is relatively modern and wide-spread programming language which is nowadays used for many tasks requiring fast development in high level language. Python is very popular, because it is user friendly (especially for beginners) and it is supported by many platforms and operating systems. Python allows procedural and object-oriented programming at the same time. Because many techniques are universal and they can be used in other programming languages, it is not hard to move from Python to another programming language.

2. Required Readings

Due to the nature of the course there is no required reading. The instructor will provide students with course reading, mostly from the Internet and resources provided directly in the operating system. Assignments for each session will be based on the reading suggested.

3. Additional Readings

- *The Python Tutorial,* by Python Software Foundation, https://docs.python.org/3/tutorial/
- *Python Programming: An Introduction to Computer Science*, by John Zelle
- *Learn Python 3 the Hard Way*, by Zed A. Shaw (Addison-Wesley, 2016)
- *Python Crash Course: A Hands-On, Project-Based Introduction to Programming*, by Eric Matthes
- *Python Cookbook: Recipes for Mastering Python 3*, by David Beazley and Brian K. Jones

4. Learning Outcomes

Upon the completion of this course, the students should be able to:

- Understand and use basic Python syntax and terminology.
- Develop basic algorithms for data manipulation.
- Understand and use essential techniques of procedural programming.
- Design an algorithmic solution for the given problem.
- Implement a program according to prepared design.
- Test and debug an implemented solution.
- Deploy the application into operating system.

5. Course Content

Week 1: Introduction to the Python language and technology, installation and introduction of a programming environment.
Week 2: Variables, data types and basic I/O (input and output).
Week 3: Lists, tuples, basic control structures (conditions and loops).
Week 4: Introduction to algorithms and flow charts.
Week 5: Structuring the solution by using functions.
Week 6: Basic data structures (stack, queue, tree).
Week 7: Error handling (exceptions).
Week 8: Advanced I/O.
Week 9: Classes.
Week 10: Python modules, deploying python applications.
Week 11: Introduction into graphical user interface.
Week 12: Reserve class, working on final assignment.
Week 13: Reserve class, working on final assignment.
Week 14: Reserve class, working on final assignment.
Week 15: Final assignment evaluation.

Since this class is more practical than theoretical, it is very hard to pass this course without periodic practice. For that reason, students will be forced to practice every week. After every lecture there will be a small in-class assignment recapitulating the main topic of the lecture. There will be also mandatory homework between lectures.
Remember that required knowledge for successful solving of all given tasks accumulates during the semester. It means that tasks will become harder during the semester and they may require the knowledge acquired in the preceding tasks.
Keep in mind that the proposed schedule may change a little depending of public holidays, the skills of the students, the current state, the progress of students' learning abilities, etc. The changes however will be minor. Also note that the order of the weeks may change, according to the teacher's and students' needs though, overall, the entire content will be provided.

6. Course Requirements and Grading

Participation	10%
Small Assignments	30%
Homework Assignments	30%
Final Assignment	30%
Total	100%

Participation in class (10%)
The instructor will mark the class activity, degree of preparedness of the students, their precise answers or intelligent questions opening discussions that lead to important knowledge discovery. Extra voluntary activity carried out by the student wanting to improve his skills related to this subject will be also taken into consideration.

Small Assignments (30%)
The instructor can assign in-class practical assignments which takes approx. from 10 to 30 minutes. This is at the latitude of the teacher and it depends on each topic taught. The number of assignments varies from semester to semester. The solution of every assignment will be revealed on the same lecture.

Homework Assignments (30%)
These assignments allow the students to practice programming skills by solving practical problems. In comparison with in-class assignments, the homework assignments are more complex since the students should finish them till the next lecture. The correct solution will be revealed during next lecture. The students will be able to test the solution automatically without lecturer interaction by using a tool given by the lecturer.

Final Assignment (30%)
The final assignment is the most complex task. Each student should design, implement and test their solution without help of a tool given by the lecturer. It means that student should not only implement the solution for the given problem, however he/she should be also able to analyze the implementation, identify all critical parts and properly test them.

Grading Scale

<table>
<tr><th>Letter Grade</th><th>Percent (%)</th><th>Generally Accepted Meaning</th><th>Notes</th></tr>
<tr><td>A</td><td>95-100</td><td rowspan="2">Outstanding work</td><td rowspan="8">Credits awarded</td></tr>
<tr><td>A-</td><td>90-94</td></tr>
<tr><td>B+</td><td>87-89</td><td rowspan="3">Good work, distinctly above the average</td></tr>
<tr><td>B</td><td>83-86</td></tr>
<tr><td>B-</td><td>80-82</td></tr>
<tr><td>C+</td><td>77-79</td><td rowspan="3">Acceptable Work</td></tr>
<tr><td>C</td><td>73-76</td></tr>
<tr><td>C-</td><td>70-72</td></tr>
<tr><td>D+</td><td>67-69</td><td rowspan="3">Work that is significantly below average</td><td rowspan="3">Credits awarded,
but will NOT transfer to ESC (retake needed for ESC!)</td></tr>
<tr><td>D</td><td>63-66</td></tr>
<tr><td>D-</td><td>60-62</td></tr>
<tr><td>F</td><td>0-59</td><td>Work that does not meet the minimum standards for passing the course</td><td>Credits not awarded</td></tr>
</table>

7. Key UNYP Policies
Attendance
It is your responsibility to show up to class on time. If you are late you will be marked as absent for that hour**.** If you miss more than 9 (nine) hours of class, for any reason, you will automatically fail the entire course. Pay strict attention to this. The class policy is standard UNYP policy.

Academic Honesty
- The university's rules on academic dishonesty (e.g., cheating, plagiarism, submitting false information) will be strictly enforced. Please familiarize yourself with the ***student honor code*** or ask your instructor for clarification.
- For examinations: copying from your neighbor, communicating with another student, using a phone or anything similar will result in you failing the test or quiz.
- On written papers, properly note your sources with academic citations. Cutting and pasting from the Internet without accurately citing the source may be considered plagiarism. Students may be required to submit papers electronically, which could mean an automated check for plagiarism via the Turn-it-in resource. Students may also be required to defend the content of a paper orally to an instructor as a check on authorship.

- If you have questions about any of the above, please consult with the instructor.

8. General Requirements

- Students are expected to attend each class session and participate in a positive way.
- Students are expected to come to class fully prepared to discuss homework readings and cases.
- Students are expected to turn in homework assignments at the beginning of the class period on the day they are due.
- Students are expected to leave their mobile phones, etc. switched off.
- Students may not use laptops or netbooks for any reason other than taking notes. Do not surf the WEB during class time. If you do, you will lose the privilege to use a laptop or netbook.
- In the event of illness or emergency, contact your instructor IN ADVANCE to determine whether special arrangements are possible.
- It is the responsibility of the student to periodically check Moodle for changes / information / details.
- Moodle is the only official mean of communication between the students and the lecturer.

9. European Credit Transfer and Accumulation System (ECTS)

The students that complete the course will receive 6 ECTS credits or 3 American credits. One ECTS credit corresponds to 25-30 hours of work. For comparison, 1 American credit hour equals approximately 2 ECTS credits.For this course, students are expected to spend time in the following course-related activities:

Class lectures and in-class assignments	24 hours
Homework assignments	50 hours
Next lecture preparation	26 hours
Final assignment	50 hours
TOTAL	150 hours

10. Technology Expectations

No previous knowledge is required. The instructor will evaluate at the beginning of the course the level of knowledge of each student in order to make balance.

Students are assumed to be familiar with the use of Internet and in particular with the gathering of data from the wide world web. This will be a necessary tool for carrying out some of the course assignments (i.e. long essay or class presentations).

There are four main goals of this subject: understanding computer basics, being able to use office tools, understanding the basics of computer networks and their problems and finally knowing how to make a research report.