

Econ 124 - Introduction to Programming for Economists**Instructor:** Kağan Parmaksız (kaganp@metu.edu.tr)**Office:** A213**Lectures:** Tue 1440-1730 – FZ16**Office Hours:** TBA

Course Description: This course is intended for students with little or no programming experience and aims to provide students with an understanding of the role computation can play in solving complex problems. It introduces core programming basics—including data types, control structures, algorithm development, and program design with functions—via the Python programming language. It covers methods such as looping and conditionals, sorting and multidimensional structures, and debugging and troubleshooting programs. By the end of this course, the students will understand the basics of computer programming, will be able to write scripts and simple programs in Python executing certain tasks of interest.

Organization of the course: The organization of the course is structured as weekly in-class lectures and applied sessions. We will mainly follow the structure of the textbook chosen for the course as we progress throughout semester. From time to time, we will use other resources and which I will announce and make them available online in advance for you to prepare for the lectures. This is a programming course and the material can only be successfully covered (learned and assessed) with a heavy emphasis on hands-on learning by doing approach.

Textbook and other useful resources: Most of the course will be based on the textbook *Think Python* 3rd Ed. by Allen B. Downey (free online version of current edition is accessible at: <https://greenteapress.com/wp/think-python-3rd-edition/>). There is an abundance of excellent texts and online material about Python programming and we will make use of them throughout the semester. When we do so, I will announce them in advance and try to make them available online.

Assessment: The assessment of this course is a little more complicated than usual, *so please read the following carefully before you commit to take this course.*

This is a computer programming course and is ought to be a course built upon week-to-week progress through use of exercises, applications and close supervision. We will assign weekly exercise sets to ensure all students in the course follows a path of incremental progress throughout the semester. To this end, the assessment of the course will be quite strict to give weekly incentive and maintain motivation of the students.

First of all, please note that *if you withdraw this course during the semester, you will not be allowed to take it again* and will need to replace it with another elective course. If you fail this course, it is very unlikely that you will be able to take it again due to capacity restrictions and will need to replace it with another elective. Last but not least, same conditions will apply for retaking the course to get a better grade, i.e. you probably won't be able to it due to capacity restrictions. Please consider all this carefully when taking the course.

The course grade will be based on a weekly assigned exercises (30%) and number of in-class exams (equally weighted, 60%) and attendance over the required amount (10%) (See below for attendance requirements). Note that assessment may cover any aspect of the course material covered in lectures, assignments and discussions made during classes. The following are the rules for grading:

1. Attendance will be taken online during **each hour of lecture**. To be able to take an exam, you are required to attend 60% of the lectures up to that exam date. Your attendance performance will be available online and it is your responsibility to keep track of this, so the requirement is *number of hours of lecture attended / number of hours of lecture attendance taken > 0.6* (do not email me or the TA about this).

2. There is absolutely no exception to the attendance requirement, medical reports and/or other emergencies will also count as unattended lecture. Coming late to class and missing the online attendance window also is not excuse and will not be fixed manually.
3. 10% of your grade will be based on the number of lectures you attend on top of the 60% required.
4. In-class exams count towards 60% of your course grade, however, *you will not get a passing grade for this course if the arithmetic average of your **in-class exams** is below 50 (out of 100).*
5. As you can guess, this last requirement is to ensure every student spend enough time on their own practicing, which is the only way to learn the coding. You are free to collaborate with classmates, use other resources and get help while doing the weekly exercises if it helps your learning process. However, you should be aware that none of these will be available during in-class exams and you need to have proficiency on the fundamentals to get a passing grade.
6. Computer programs written to execute a certain task have a binary outcome, i.e. they either work or don't. Even the best coders of the world rarely get it at once. The best way to learn programming is accumulating knowledge through your mistakes and the only way to accomplish that is by making mistakes AND figuring them out on your own to fix them. For weekly exercises, the grading will be based on whether your code works and is written along the intended learning outcomes or not. You will have a chance to check it and fix it before you submit.
7. During in class-exams however, you will be graded mostly based on your understanding of the fundamental material, the structure of your code, use of good coding practices etc. This level of understanding of the material is ONLY possible if you spend adequate time on the weekly exercises.

Communication: <https://odtuclass.metu.edu.tr>

Note that all information regarding the course, reading materials, assignments, exams and announcements will go through this system. It is your responsibility to check this site at regular intervals and be up-to-date with all relevant information.

Academic Honesty: Any form of dishonest behavior for any part of the course, which includes but not limited to; cheating during exams, plagiarizing and inappropriate collaboration in assignments, will ensure a failing grade for the course and will result in further disciplinary action in line with university regulations depending on the severity of the violation.

Information for Students with Disabilities Students who experience difficulties due to their disabilities and wish to obtain academic adjustments and/or auxiliary aids must contact ODTU Disability Support Office and/or course instructor and the advisor of students with disabilities at academic departments (for the list: <http://engelsiz.metu.edu.tr/en/advisor-students-disabilities>) as soon as possible. For detailed information, please visit the website of Disability Support Office: <https://engelsiz.metu.edu.tr/en/>

Course Outline: The brief outline for the course can be seen at the site of online version of our textbook

<https://allendowney.github.io/ThinkPython/>

The detailed reading assignments will be made available in advance before the lecture on course website each week.