

COMP125: Programming with Python (Spring 2021)

Course Description

This course emphasizes computational thinking and programming using Python. Students are expected to gain a solid foundation in algorithmic thinking and structured programming, and perform basic, common computational tasks easily and efficiently. The course examines the fundamentals of storage, input and output, control structures, functions, sequences and lists, file I/O, numerical computation, visualization and object-oriented programming.

The course will be offered in two sections in parallel.
(Section 1) – Monday, Wednesday, 1.00 – 2.15 PM
(Section 2) – Monday, Wednesday, 4.00 – 5.15 PM

Instructor:

Beren Semiz, besemiz@ku.edu.tr

Online office hours: Wednesday 5.30 – 6.30 PM

Course Textbook

[Starting out with Python](#), 4th edition, by Tony Gaddis, Pearson Addison-Wesley

KU Credits: 3.00

ECTS Credits: 6.00

Language: English

Lab Schedule:

(Lab A) – Friday, 8.00 – 9.50 AM

(Lab B) – Friday, 10.00 – 11.50 AM

(Lab C) – Friday, 4.00 – 5.50 PM

(Lab D) – Friday, 6.00 – 7.50 PM

Teaching Assistants (TAs):

TBA

Section Leaders (SLs):

TBA

Course Learning Outcomes (CLOs)

- Gain a solid foundation in algorithmic thinking and structured programming.
- Perform basic, common computational tasks easily and efficiently.
- Examine fundamentals of data storage, input and output, control structures, functions, sequence and lists, file I/O, numerical computation, visualization and object-oriented programming.

Online Teaching/Learning

- Course will be taught online. Course communication will be through Blackboard, e-mail, online face-to-face meetings, online office hours, and Blackboard Discussion Board.
- **Lectures:** Online face-to-face lectures using Zoom. Zoom links will be available on Blackboard. Lectures will be automatically recorded by Panopto and recordings will be made available after the lecture.
- **Labs:** There will be a lab session every week. Labs will be dedicated to collaborative coding exercises. Students will receive a lab participation grade.

Assessment Method

- Lab Participation (10%): There will tentatively be 12 labs. You are expected to attend at least 10 lab sessions.
- Homework (20%): There will tentatively be 4 graded homework assignments.
- Midterms (45%): There will tentatively be 4 midterms. Lowest midterm grade will be dropped when calculating your midterm average.
- Final exam (25%): Comprehensive programming exam to be scheduled during the final week

There will not be make-ups for midterms and homeworks. Final exam make-up will be scheduled by the registrar's office.

Late Submission Policy

Late homework submissions, i.e., homework submitted after its deadline, will be penalized.

Student's late homework grade (late_HW_grade) will be calculated as:

$$\text{late_HW_grade} = \text{HW_grade} * \left(1 - \frac{x}{6000}\right)$$

where HW_grade is the grade the student would have received if they had submitted on time, and x is the time beyond the deadline in minutes. Late submissions are allowed only until 48 hours after the deadline; after that, the grade will automatically be 0.

TA/SL Office Hour Schedule

TAs and SLs will hold online office hours every week. You are encouraged to attend their office hours to get one-on-one help. Office hour schedules will be announced on Blackboard after the semester starts.

Tools and Infrastructure

We will use Anaconda, which comes with Python, popular Python packages, and development environments such as Spyder and Jupyter Notebook. You can download and install Anaconda from: <https://www.anaconda.com/products/individual> depending on your operating system (Windows, Linux, or MacOS).

Please download Anaconda and familiarize yourself with Python and Spyder as soon as possible.

Tentative Schedule

- Welcome and introduction
- Designing and developing a program
- Variables, expressions, control flow
- Loops
- Functions
- Strings and string operations
- Data structures: lists, dictionaries, sets, and tuples
- File I/O
- Data parsing
- Nested data structures
- Exceptions and exception handling
- Recursion
- Numpy: arrays, functions, linear algebra & engineering applications
- Matplotlib: plotting and visualization
- Object-oriented programming
- Inheritance

All students are expected to comply with the [Koç University Student Code of Conduct \(https://apdd.ku.edu.tr/en/academic-policies/student-code-of-conduct/\)](https://apdd.ku.edu.tr/en/academic-policies/student-code-of-conduct/). Cheating will not be tolerated. All homework and exam submissions will be checked for similarity and plagiarism.