

<b>Identification</b>	<b>Subject</b>	CMS140, Fundamentals of Computer Programming, 6 ECTS	
	<b>Department</b>	Computer Science	
	<b>Program</b>	Bachelor	
	<b>Term</b>	Spring 2026	
	<b>Instructor</b>	Humay Huseynova	
	<b>Email:</b>	hhuseynova@khazar.org	
	<b>Classroom/hours</b>	401N, Thursday, 15:20-17:00	
<b>Prerequisites</b>	CM 115: Computer Applications in Engineering		
<b>Language</b>	English		
<b>Compulsory/Elective</b>	Compulsory		
<b>Textbooks and course materials</b>	<p>Core textbooks:</p> <ol style="list-style-type: none"> <li>1. Python Basics: A Practical Introduction to Python 3. David Amos, Dan Bader, Joanna Jablonski, Fletcher Heisler, 2022</li> <li>2. Python QuickStart Guide. Robert Oliver, 2023</li> <li>3. An introduction to Programming using Python. David I. Schneider, 2019</li> <li>4. Fundamentals of Python Programming by Richard L. Halterman, 2018</li> </ol>		
<b>Course description</b>	<p>This course is designed to provide students with a foundation in computer programming. Throughout this course, students will learn the fundamentals of programming. Starting with an overview of the importance of programming languages, the course progresses to cover topics such as data types and variables, data structures, functions, handling errors, reading files etc.</p>		
<b>Course objectives</b>	<ul style="list-style-type: none"> <li>• Appreciate the power of programming languages: This emphasizes the problem-solving capabilities and automation potential that programming languages offer.</li> <li>• Demystify structured programming: Explore its concepts and practical applications: This clarifies the focus on understanding and using structured programming techniques.</li> <li>• Applying Python programming for building real-world software: This highlights the goal of gaining practical skills in Python language to create functional applications</li> </ul>		
<b>Learning Outcomes</b>	<p>Upon completion of this course, the students must be able to:</p> <ul style="list-style-type: none"> <li>– Understand the basics of programming languages, like variables, data types, operators, and control flow.</li> <li>– Utilize various programming structures effectively, such as if-else statements, loops (for, while), and functions.</li> <li>– Implement common algorithms and data structures in code.</li> <li>– Analyze and Solve Programming Problems.</li> <li>– Structure programs with clear logic and flow control.</li> </ul>		
<b>Teaching methods</b>	<b>Lab</b>		X
	<b>Group discussion</b>		X
	<b>Lecture</b>		X
	<b>Simulation</b>		X

Evaluation Criteria	Methods	Date/deadlines	Percentage (%)
	<b>Midterm Exam</b>	TBA	30
	<b>Quizzes</b>	March and April	10
	<b>Practical Assignments</b>	May	5
	<b>Individual project</b>	May	10
	<b>Attendance</b>	During semester	5
	<b>Final Exam</b>	TBA	40
	<b>Total</b>		100
<b>Class Policy</b>	<p><b>Classroom activities description</b></p> <p><b>Attendance (5 points)</b> Students who attend all classes will receive the full 5 points. For every <b>2</b> missed classes, <b>1</b> point will be deducted.</p> <p><b>Practical Assignments (5 points)</b> Students can earn up to 5 points through in-class discussions and home assignments. Some tasks will be completed in class, while others will be assigned as homework. To receive full points, all assignments must be submitted on time.</p> <p><b>Quizzes (10 points total)</b> There will be 2 quizzes throughout the semester: Quiz 1: March Quiz 2: April Each quiz is worth 5 points. Students who miss a quiz without a valid reason will not be allowed a retake.</p> <p><b>Individual Project</b> Task: Students will be given a real-world problem to solve using Python. Evaluation: Code Quality (5 points): Clarity, efficiency, correctness, readability, adherence to best practices. Explanation (5 points): Understanding of the problem, logic, and reasoning behind the approach. Submission: Students must submit their code until Week 13 and present their solutions in Week 14</p> <p><b>Preparation for class</b> The lecture material will focus on the major points introduced in the text. Reading the assigned chapters and having some familiarity with them before class will greatly assist your understanding of the lecture.</p> <p><b>Withdrawal (pass/fail)</b> This course strictly follows the grading policy of the School of Engineering and Applied Science. Thus, a student is normally expected to achieve a mark of at least 60% to pass. In case of failure, he/she will be required to repeat the course the following term or year.</p> <p><b>Cheating/plagiarism</b></p>		

	<p>Cheating or other plagiarism during the Quizzes, Mid-term and Final Examinations will lead to paper cancellation. In this case, the student will automatically get zero (0), without any consideration.</p> <p><b>Professional behavior guidelines</b></p> <p>The students shall behave in a way to create a favorable academic and professional environment during class hours. Unauthorized discussions and unethical behavior are strictly prohibited.</p> <p><b>Ethics</b></p> <p>Students should not arrive late at class. All cell phones must be turned off or silent and before entering class. Use of electronic devices is not allowed in the classroom, and violators will be punished accordingly</p>
--	--

**Tentative Schedule**

<b>Week</b>	<b>Date/Day (tentative)</b>	<b>Topics</b>	<b>Textbook/Assignments</b>
1	19.02.26	Introduction to Programming & Python Overview of programming languages, Python installation and setup, IDEs. Introduction to the Python environment	Ch.1 Introduction to Programming using Python by David I. Schneider  Introduction - Automate the Boring Stuff with Python by Al Sweigart  Lab: Write a "Hello, World!" program
2	26.02.26	Variables, Data Types, and Expressions Introduction to variables, basic data types (int, float, string), arithmetic operations	Ch. 2 Introduction to Programming using Python by David I. Schneider  Python basics - Automate the Boring Stuff with Python by Al Sweigart  Lab: Create simple arithmetic programs using variables
3	05.03.26	Conditional Statements Boolean expressions, if/else statements, nested conditions.	Ch. 4 Fundamentals of Python Programming by Richard L. Halterman,

			<p>Flow control - Automate the Boring Stuff with Python by Al Sweigart</p> <p>Lab: Build a basic calculator using if/else statements</p>
4	12.03.26	Loops: While and For Introduction to loops, while loops, for loops, loop control (break, continue).	<p>Ch 3 Introduction to Programming using Python by David I. Schneider</p> <p>Flow control - Automate the Boring Stuff with Python by Al Sweigart</p> <p>Lab: Write programs that print numbers, calculate factorials, etc.</p>
5	19.03.26	Functions Defining functions, return values, parameters, function scope. Quiz 1	<p>Ch. 4-Introduction to Programming using Python by David I. Schneider</p> <p>Functions - Automate the Boring Stuff with Python by Al Sweigart</p> <p>Lab: Write functions to solve basic problems</p>
6	26.03.26	Non-working day	
7	02.04.26	Lists and Tuples Introduction to lists and tuples, list indexing, slicing, and list operations.	<p>Ch.10- Fundamentals of Python Programming by Richard L. Halterman,</p> <p>List - Automate the Boring Stuff with Python by Al Sweigart</p> <p>Lab: Create a shopping list Program and practice list manipulations</p>
8	09.04.26	<b>Midterm exam</b>	

9	16.04.26	Dictionaries and Sets Understanding dictionaries and sets, creating key-value pairs, set operations.	<p>Ch. 11 Fundamentals of Python Programming by Richard L. Halterman,</p> <p>Dictionaries - Automate the Boring Stuff with Python by Al Sweigart</p> <p>Lab: Write a program that counts word frequency in a sentence</p> <p>Project Discussion &amp; Guidelines</p>
10	23.04.26	File Handling. Reading and writing text files, file operations.	<p>Presentation</p> <p>Reading and writing files - Automate the Boring Stuff with Python by Al Sweigart</p> <p>Lab: Build a program that stores and retrieves data from files</p>
11	30.04.26	Error Handling Introduction to exception handling, try/except blocks. Quiz 2	<p>Ch. 12 Fundamentals of Python Programming by Richard L. Halterman,</p> <p>Lab: Write a program that handles user input errors</p>
12	07.05.26	More on functions. Global variables. Lambda expressions. Decorators	Ch.8 Fundamentals of Python Programming by Richard L. Halterman,
13	14.05.26	Modules and Libraries Importing built-in modules, third-party libraries.	<p>Presentation</p> <p>Lab: Use Python's random and math modules to create a simple math quiz</p>
14	21.05.26	Project Submission and Presentations Students submit and present their projects. Projects are evaluated based on code quality and explanation.	

15	28.05.26	Non-working day	
		<b>Final exam</b>	