REPUBLIKA E SHQIPËRISË

"ISMAIL QEMALI" UNIVERSITY
FACULTY OF TECHNICAL AND NATURAL SCIENCES
COMPUTER SCIENCE DEPARTMENT

SYLLABUS
Structure and interpretation of computer programs

Curriculum Developer		**PhD. Dezdemona Gjylapi**							
Credits and hours	Credits	Hour in the classroom						Individual study	Total
		Lectures	Seminars	Labs	Practice	Course project	Total		
	8 ECTS	45	30	15	0	0	90	110	200
Course Typology		C - Interdisciplinary/Integrative disciplines							
Academic year /semester		2025-2026/ First year, first semester							
Nature and requirement of the course		Mandatory course							
Study program		Bachelor in Computer Engineering							
Course Code		CS 111E							
Email of curriculum developer / lecturer		dezdemona.gjylapi@univlora.edu.al							

SUMMARY AND LEARNING OUTCOMES

Course Description	This course provides a rigorous introduction to the principles of programming and computer science, emphasizing the importance of abstraction, modularity, recursion, and higher-order functions. The course uses Python 3 as a programming language to teach fundamental concepts that transcend specific programming languages. Through a series of lectures, problem sets, and projects, students will explore a variety of computational models, including procedural, functional, and declarative paradigms. They will learn to design powerful and elegant software systems by building interpreters, implementing object-oriented models, and constructing domain-specific languages. The course also covers essential concepts such as data abstraction, environment models, stateful computations, and metalinguistic abstraction.
Course Objectives	▪ To develop problem -solving skills. ▪ To understand and apply abstraction. ▪ To implement recursive and iterative processes and analyze their efficiency. ▪ To design systems that use assignment and state to manage complex processes.
Expected Outcomes	Upon successful completion of the course, students will be able to: ● Design programs using various levels of abstraction, including procedural, data, and metalinguistic abstractions.

	● Implement higher-order functions, recursive processes, and analyze computational efficiency. ● Construct modular software systems that leverage object-oriented principles and state management. ● Build simple interpreters and understand the principles behind language design and evaluation. ● Apply functional and logical programming paradigms to solve complex problems. ● Develop robust problem-solving skills applicable to various programming languages and paradigms.

Week	COURSE TOPICS	Reference Literature	Teaching Methodology
I.	***Topic 1*** *– Introduction to the CS 111E Syllabus. Building Abstractions with functions.Installing Python 3.Programming in Python.Interactive Sessions.First Example.Errors.* ***Topic2****– Elements of Programming.Expressions.Call Expressions. Importing Library Functions.* ***Topic 3*** *–Names and the Environment.Evaluating Nested Expressions.The Non-Pure Print Function* ***Seminar -*** *Elements of Programming. Importing Library Functions*	*Syllabys CS 111E* *[1] page. 29-62*	Lecture 1 hour Lecture 1 hour Lecture 1 hour Seminars 2 hours Lab 1 hour
II.	***Topic 4*** *– Defining new Functions.Environment.Calling user-defined functions.* ***Topic 5*** *– Local Names.Choosing names.* ***Topic 6*** *– Functions as Abstractions.Operators* ***Seminar*** *-.Functions. Operators.Local Names*	*[1] page. 63-87*	Lecture 1 hour Lecture 1 hour Lecture 1 hour *Seminars 2 hours* Lab 1 hour
III.	***Topic 7*** *–Control.Statements.Compound Statements.Defining Functions II:Local Assignment* ***Topic 8*** *–.Conditional Statements.Iteration* ***Topic 9*** *–.Testing* ***Seminar*** *– Exercise in Python . Conditional Statements.Iteration*	*[1] page. 88-103* *[2] page 159-192*	Leksion 1 orë Leksion 1 orë Leksion 1 orë *Seminare 2 orë* Lab 1 hour
IV.	***Topic 10*** *– Higher order functionsFunctions as ArgumentsFunctions as GeneralMethods.* ***Topic11-*** *Defining Functions III:Nested Definitions.Functions as returned values.Example:Newton's Method* ***Topic 12*** *– Currying.Lambda Expressions.* ***Seminar*** *– Functions*	*[1] page.103-124*	Lecture 1 hour Lecture 1 hour Lecture 1 hour *Seminars 2 hours* Lab 1 hour

REPUBLIKA E SHQIPËRISË

"ISMAIL QEMALI" UNIVERSITY

FACULTY OF TECHNICAL AND NATURAL SCIENCES

COMPUTER SCIENCE DEPARTMENT

Week	COURSE TOPICS	Reference Literature	Teaching Methodology
V.	***Topic 13*** *–The Anatomy of Recursive Functions.Mutual Recursion.* ***Topic 14*** *–Printing in Recursive Functions* ***Topic 15*** *–Tree Recursion.Example:Partitions.* ***Seminar*** *– Tree Recursion*	*[1] page. 124-134*	Lecture 1 hour Lecture 1 hour Lecture 1 hour *Seminars 2 hours* Lab 1 hour
VI.	***Topic 16*** *– Native Data Types.Data Abstractions.Rational Numbers.Pairs.Abstraction Barriers.The properties of Data* ***Topic 17*** *–Lists.Sequence Iteration.Sequence Processing.Sequence Abstractions* ***Topic 18*** *–Strings. Trees.Linked Lists* ***Seminar -*** *Exercise in Python with lists,strings,linked lists,trees*	*[1] page. 135-184* *[2] page 9-113*	Lecture 1 hour Lecture 1 hour Lecture 1 hour *Seminars 2 hours* Lab 1 hour
VII	***Topic 19 –****Implicit Sequences.Iterators.Iterables.Built-in iterators.* ***Topic 20 –****For Statements.Generators and Yield Statements.Iterable Interface* **Topic 21 –***Creating iterables with yields.Iterator Interface.Streams.Python Streams.* ***Seminar –*** *Iterators. For Statements. Python Streams*	*[1] page. 135-184 & 294-479*	Lecture 1 hour Lecture 1 hour Lecture 1 hour *Seminars 2 hours* Lab 1 hour
VIII.	***Topic 22*** *–Objects-Oriented Programming.Objects and classes.Defining classes.Message Passing and Dot Expressions* ***Topic 23*** *–Class Attributes .Inheritance.Using Inheritance.* ***Topic 24*** *–Multiple Inheritance.The role of objects* ***Seminar -*** *Objects and classes. Class Attributes*	*[2] page. 234-250*	Lecture 1 hour Lecture 1 hour Lecture 1 hour *Seminars 2 hours* Lab 1 hour
IX.	***Topic 25*** *–Implementing Classes and Objects.Instances.Classes.Using Implemented Objects.* ***Topic 26*** *–Object Abstraction.String Conversion.Special Methods.Generic Functions.* ***Topic 27*** *–Efficiency.Memorization.Orders of Growth. Growth Categories* ***Seminar -*** *Implementing Classes and Objects.Exercise for orders of growth.*	*[2] page. 251-276*	Lecture 1 hour Lecture 1 hour Lecture 1 hour *Seminars 2 hours* Lab 1 hour
X.	***Topic 28*** *–.File handling.Writing and Reading Binary Data.* ***Topic 29*** *–Writing and Parsing Text Files* ***Topic 30*** *–Writing and Parsing XML files . Random Access Binary Files* ***Seminar -*** *File handling. Writing and Parsing Text Files*	*[2] page.287-332*	Lecture 1 hour Lecture 1 hour Lecture 1 hour *Seminars 2 hours* Lab 1 hour

REPUBLIKA E SHQIPËRISË

"ISMAIL QEMALI" UNIVERSITY

FACULTY OF TECHNICAL AND NATURAL SCIENCES

COMPUTER SCIENCE DEPARTMENT

Week	COURSE TOPICS	Reference Literature	Teaching Methodology
XI.	***Topic 31*** *–Interpreting Computer Programs . Programming Languages Scheme.Functional Programming.Expressions.Definitions.* ***Topic 32*** *–Compound values. Symbolic Data.Turtle graphics* ***Topic 33*** *– Exception objects* ***Seminar-*** *Programming Languages Scheme.Functional Programming.Expressions.Definitions*	*[1] page.*	Lecture 1 hour Lecture 1 hour Lecture 1 hour *Seminars 2 hours* Lab 1 hour
XII.	***Topic 34*** *–Interpreters for Languages with Combination. A Scheme-Syntax Calculator.Expression Trees* ***Topic 35*** *–Parsing Expressions.* ***Topic 36*** *– Calculator Evaluation* ***Seminar*** *-. A Scheme-Syntax Calculator.Expression Trees*	*[1] page. 492-555*	Lecture 1 hour Lecture 1 hour Lecture 1 hour *Seminars 2 hours* Lab 1 hour
XIII.	***Topic 37*** *– Interpreters for Languages with Abstraction.Structure* ***Topic 38*** *–Environments.* ***Topic 39*** *– Data as Programs* ***Seminar*** *- Data as Programs*	*[1] page. 556-659*	Lecture 1 hour Lecture 1 hour Lecture 1 hour *Seminars 2 hours* Lab 1 hour
XIV.	***Topic 40*** *–.SQL.Data types* ***Topic 41*** *–Tables* ***Topic 42*** *–Functions(Aggregate)* ***Seminar*** *– Exercise in Sql*	*[2] page 475-488* *[3] page 8-98*	Lecture 1 hour Lecture 1 hour Lecture 1 hour *Seminars 2 hours* Lab 1 hour
XV.	***Topic 43*** *–Databases.Sequence* ***Topic 44*** *– Subqueries* ***Topic 45*** *– Stored Procedures.Triggers* ***Seminar*** *– Exercise in Sql*	*[3] page.114-140*	Lecture 1 hour Lecture 1 hour Lecture 1 hour *Seminars 2 hours* Lab 1 hour

REPUBLIKA E SHQIPËRISË

"ISMAIL QEMALI" UNIVERSITY
FACULTY OF TECHNICAL AND NATURAL SCIENCES
COMPUTER SCIENCE DEPARTMENT

KNOWLEDGE ASSESSMENT

ATTENDANCE

Attendance for lectures and seminars is mandatory at 75%. Otherwise, the student loses the right to take the exam. Absences are the personal responsibility of the student, as are any missed assignments or tests.
A student who has less than 75% attendance for the period relevant to any partial exam, during which they will be tested, will not be allowed to take the corresponding exam and will be graded with an M (Missing). If the student has attended the course but does not show up for the next exam, they will be graded NP (Not Present).

CONTINUOUS ASSESSMENT

Assessment Method	*Assessment Time*	*Learning Topics to be Assessed*	*Maximum Points*	*Knowledge Testing Method*	*Test Content*
Partial Exam	Week 7-8	1-15	25 points	Written exam	Concepts and exercises
Final Exam	End of the semester	16-45	60 points	Written exam	Concepts and exercises
Continuous Assessment* (Course Assignment*)	During the semester	-	15 points	Written report and oral presentation	*Create a simplified interpreter*

* Continuous Assessment – *Create a simplified interpreter for a subset of the Scheme programming language using Python .*

The conversion of points to the final grade is as follows:

0 - 40 points	41 - 50 points	51 - 60 points	61 - 70 points	71 - 80 points	81 - 90 points	91 - 100 points
4	5	6	7	8	9	10

LITERATURE

Primary literature	[1] Structure and Interpretation of Computer Programs - 2nd Edition by Harold Abelson (Author), Gerald Jay Sussman (Author), Julie Sussman (Contributor) ISBN-10 0262510871
Recommended literature	[2] Programming in Python 3. A Complete Introduction to the Python Language Second Edition Mark Summerfield ISBN-13: 978-0-321-68056-3 [3]SQLNotesForProfessionals [4] Composing Programs by John DeNero https://cs61a.org/

REPUBLIKA E SHQIPËRISË

"ISMAIL QEMALI" UNIVERSITY
FACULTY OF TECHNICAL AND NATURAL SCIENCES
COMPUTER SCIENCE DEPARTMENT

FINAL REMARKS FROM THE COURSE INSTRUCTOR	
Provisions	Students are expected to develop values and an ethics that reflect their interests and behavior from a professional perspective. Students must respect individual differences, be cooperative, and in accordance with ethical rules, they should respect ethnic differences and assist individuals with disabilities with inclusive and supportive attitudes.
Delays	Delays in lectures and seminars will not be tolerated.
Academic integrity and copyright	Each test or written work must be completely individual and original. Plagiarism (mixed combinations) or theft of information through various means will not be allowed.

HEAD OF DEPARTMENT OF COMPUTER SCIENCES
Dr. Alketa HYSO