Course Title:	**Functional programming**
Course Code:	CSE451/1
Program:	Master Degree In Computer Engineering
Department:	Computer Engineering
Course coordinator:	Dr. Mounir TELLI
Institution:	Private Higher School of Engineers of Gafsa (ESIP)

A. Course Identification

1. Credit hours: 3 (1.5-0-1.5)
2. Course type **a.** College ☐ Department ■ Others ☐ **b.** Fundamental ■ Transversal ☐ Optional ☐
3. Level/year at which this course is offered: 2.2/3
4. Pre-requisites for this course (if any): CSE131, CSE132
5. Co-requisites for this course (if any)**:**

1. **Mode of Instruction** (mark all that apply)

No	Mode of Instruction	Contact Hours	Self-study	Total workload
1	**Traditional classroom**	…..	33	78
2	**Blended**	45		
3	**E-learning**	.....		
4	**Distance learning**	.....		
5	**Other ()**	.....		

2. **Contact Hours** (based on academic semester)

No	Activity	Contact Hours
1	**Lecture**	22.5
2	**Laboratory/Studio**	22.5
3	**Tutorial**	-
4	**Others** (specify)	-
	Total	45

B. Course Objectives and Learning Outcomes

Course Description

This course introduces functional programming in Python, focusing on writing clean, efficient, and reusable code. Students will learn key concepts like higher-order functions, recursion, lambda expressions, and immutable data types**.**

The course covers functional tools such as map, filter, and reduce, as well as list comprehensions and generator expressions. Students will also explore advanced topics like memoization, tail recursion, and data stream processing**.**

By the end of the course, students will be able to use functional programming techniques to solve real-world problems in Python and improve their coding skills.

Course Main Objective

- Understand basic functional programming concepts and principles, such as higher-order functions, lambda expressions, recursion, function composition, and more.
- Understand how functional programming differs from other programming paradigms, such as imperative programming and object-oriented programming.
- Learn to use functional programming tools in Python, such as map, filter, and reduce functions, lambda expression syntax, generators, and list comprehensions.
- Learn how to create functional programs in Python, using techniques such as pure functional programming, data immutability, and recursion.
- Understand how functional programming can improve the readability, maintainability, and performance of Python code.
- Discover how functional programming can be used in applications such as data manipulation, signal processing, machine learning, and numerical analysis.
- Gain hands-on experience programming real-world examples in Python, such as creating programs to manipulate lists, arrays, and other data structures, as well as sample applications such as data analysis and machine learning.

1. Course Learning Outcomes

CLOs		Aligned PLOs
	Knowledge and Understanding	
1.2	- Understand basic functional programming concepts and principles, such as higher-order functions, lambda expressions, recursion, function composition, and more.	PLO.K1
1.2	- Understand how functional programming differs from other programming paradigms, such as imperative programming and object-oriented programming.	
1.3	- Learn to use functional programming tools in Python, such as map, filter, and reduce functions, lambda expression syntax, generators, and list comprehensions.	
1.4	- Learn how to create functional programs in Python, using techniques such as pure functional programming, data immutability, and recursion.	
1.5	- Understand how functional programming can improve the readability, maintainability, and performance of Python code.	
	SKILLS	
1.1	Gain hands-on experience programming real-world examples in Python, such as creating programs to manipulate lists, arrays, and other data structures, as well as sample applications such as data analysis and machine learning.	PLO.S1
7.1	Promote effective collaboration and communication among multidisciplinary teams to achieve software design and development objectives through function programing.	PLO.S7

C. Course Content

No	List of Topics	Contact Hours
1	❖ Introduction to Functional Programming in Python - Definitions and basic principles - Comparison with other programming paradigms - Advantages and Disadvantages of Functional Programming in Python	3
2	❖ Functions - Functions as values - Higher Order Functions - Recursive functions - Lambda Expressions - Partial functions	6
3	❖ Immutable data types - Character strings	6

	- Tuples - Sets - Immutable Dictionaries	
4	❖ Lists in Python - Lists as sequences - Manipulating Lists with Map, Filter, and Reduce Functions - List comprehensions and generator expressions - List reduction with lambda functions	6
5	❖ Higher order functions - Introduction to Higher Order Functions - Using Higher Order Functions as Arguments and Results - Practical Examples of Using Higher Order Functions in Python	6
6	❖ Advanced recursive functions - Analysis of more complex problems solved by recursive functions - Using advanced techniques like memoization and tail recursion	6
7	❖ Manipulation of data streams - Introduction to Data Flows - Using data flow manipulation functions (map, filter, reduce) - Practical Examples of Using Data Flow Manipulation in Python	6
8	❖ Practical applications of functional programming - Examination of real use cases of functional programming - Completion of a hands-on project implementing the concepts and techniques learned	6
Total		45

D. Teaching and Assessment

1. Alignment of Course Learning Outcomes with Teaching Strategies and Assessment Methods

Code	Course Learning Outcomes	Teaching Strategies	Assessment Methods
1.0	**Knowledge and Understanding**		
	- Understand basic functional programming concepts and principles, such as higher-order functions, lambda expressions, recursion, function composition, and more. ✓ Understand how functional programming differs from other programming paradigms, such as imperative programming and object-oriented programming.	Lecturing	Assignments, Quizzes, Exams,

Code	Course Learning Outcomes	Teaching Strategies	Assessment Methods
	✓ Learn to use functional programming tools in Python, such as map, filter, and reduce functions, lambda expression syntax, generators, and list comprehensions. ✓ Learn how to create functional programs in Python, using techniques such as pure functional programming, data immutability, and recursion. ✓ Understand how functional programming can improve the readability, maintainability, and performance of Python code. ✓ Discover how functional programming can be used in applications such as data manipulation, signal processing, machine learning, and numerical analysis.		
3.0	**Skills**		
PLO.S1	✓ Gain hands-on experience programming real-world examples in Python, such as creating programs to manipulate lists, arrays, and other data structures, as well as sample applications such as data analysis and machine learning	- Lectures - Class discussions - Assignments - projects	Assignments, Report, Quizzes, Exams
PLO.S7	✓ Promote effective collaboration and communication among multidisciplinary teams to achieve software design and development objectives through function programing.		

2. Assessment Tasks for Students

#	Assessment task*	Week Due	Percentage of Total Assessment Score
1	Practical Work (written or oral)	Weekly	25%
2	Quizzes, Homework assignments	Random	00%
3	First mid Term	8	25%
4	Final Exam	16	50%

E. Student Academic Counselling and Support

Arrangements for availability of faculty and teaching staff for individual student consultations and academic advice:
- Office hours - Blackboard interface - Academic advisor - Bibliotic

F. Learning Resources and Facilities

1. Learning Resources

Required Textbooks	**1.** Functional Programming in Python, David Mertz, May 2015, O'Reilly Media, Inc. **2.** Functional Python Programming, Steven F. Lott, April 2018, Packt. **3.** Python for Data Analysis, by William McKinney, November 2017, O'Reilly Media, Inc. **4.** Python Cookbook, 3rd Edition, David Beazley, Brian K. Jones, Released May 2013, O'Reilly Media, Inc.
Essential References Materials	- NA
Electronic Materials	1. Python Official Documentation: Functional Programming Concepts in Python (docs.python.org) 2. MIT OpenCourseWare: Functional Programming with Python (ocw.mit.edu) 3. Coursera: Python Functional Programming Courses from top universities (coursera.org)
Other Learning Materials	**1.** Python Libraries: NumPy, Pandas, and functools for functional programming techniques **2.** GitHub Repositories: Code examples of real-world functional programming applications

2. Facilities Required

Item	**Resources**
Accommodation	**Classroom board** **Computer lab with the necessary software** **Internet access**
Technology Resources	**Data projector**

G. Course Quality Evaluation

Evaluation Areas/Issues	Evaluators	Evaluation Methods
Effectiveness of teaching and assessment.	Students, course coordinator, Alumni, Employers	Direct/Indirect
Extent of achievement of course learning outcomes.	Faculty, Program Leaders, quality department	Direct
Quality of Learning resources	Faculty, Program Leaders,	Direct, Indirect
Teaching and learning quality and effectiveness.	Students, Faculty Program Leaders,	Direct, Indirect

H. Specification Approval Data

Council / Committee	Computer Engineering Council
Date	07/02/2024