



Course Specification

DIPLOMA

Course Title: **Fundamentals of Algorithms and Programming**

Course Code: **APDA1204**

Program: **The Data Analytics Diploma**

Department: **Diploma Department**

College: **The Applied College**

Institution: **Umm Al-Qura University**

Version: **1**

Last Revision Date: **05 May 2025**



Table of Contents

A. General information about the course:	3
B. Course Learning Outcomes (CLOs), Teaching Strategies and Assessment Methods	4
C. Course Content.....	5
D. Students Assessment Activities	7
E. Learning Resources and Facilities.....	7
F. Assessment of Course Quality	8
G. Specification Approval	8



A. General information about the course:

1. Course Identification

1. Credit hours: (3)

2. Course type

A. ☐ University ☐ College ☒ Department ☐ Track ☐ Others
B. ☒ Required ☐ Elective

3. Level/year at which this course is offered: (level 2, 1st year)

4. Course general Description:

This course introduces the basic concepts of computer programming to students with some problem-solving skills. Students will be using Python, a high-level programming language, to learn the fundamentals of computer programming including how to write, compile, and run programs in an IDE. Topics include algorithms and problem solving, variables and data types, methods, console input/output, control structures, arrays, Python best coding styles, and the mechanics of running, testing, and debugging.

5. Pre-requirements for this course (if any):

None

6. Pre-requirements for this course (if any):

None

7. Course Main Objective(s):

To equip students with the fundamental knowledge and skill to solve computational problems using Python programming language.

2. Teaching mode (mark all that apply)

No	Mode of Instruction	Contact Hours	Percentage
1	Traditional classroom	60	100%
2	E-learning		
3	Hybrid <ul style="list-style-type: none"> Traditional classroom E-learning 		
4	Distance learning		



3. Contact Hours (based on the academic semester)

No	Activity	Contact Hours
1.	Lectures	30
2.	Laboratory/Studio	30
3.	Field	
4.	Tutorial	
5.	Others (specify)	
Total		60

B. Course Learning Outcomes (CLOs), Teaching Strategies and Assessment Methods

Code	Course Learning Outcomes	Code of CLOs aligned with program	Teaching Strategies	Assessment Methods
1.0	Knowledge and understanding			
1.1	Basic understanding of Python programming language	K1	Course lectures, project	Quizzes, Midterm Exam, Final Exam
1.2	Understand and use basic control structures	K1	Course lectures, lab exercises, project	Quizzes, Midterm Exam, Final Exam
2.0	Skills			
2.1	Write Python programs that solve simple problems	S1	Lab coursework Project	Quizzes, Midterm Exam, Final Exam, project
2.2	Design simple programs using computing structures such as loops, conditions, functions, arrays and OOP	S1	Lab coursework Project	Quizzes, Midterm Exam, Final Exam, project
2.3	Using computer machines and software tools to solve computer problems	S1	Lab coursework Project	Quizzes, Midterm Exam, Final Exam, project
3.0	Values, autonomy, and responsibility			
3.1	Work in groups	V2	Project	Project
3.2	Be an independent learner, able to acquire further	V1, V2	Lecture	Assignments, Quizzes, Exams





Code	Course Learning Outcomes	Code of CLOs aligned with program	Teaching Strategies	Assessment Methods
	knowledge with some guidance or support.			

C. Course Content

No	List of Topics	Contact Hours	
		Lecture	Lab
1.	Introduction to Computer Science and Programming in Python: <ul style="list-style-type: none"> Introduction to computational thinking and problem solving Introduction to Python and its ecosystem Mathematical Operations Basic syntax: variables, keywords, and expressions Data types: integers, floats, and strings Basic input/output (print, input) Lab: <ul style="list-style-type: none"> Install Python and a development environment (e.g., Anaconda or Jupyter Notebook) Write simple Python scripts: "Hello, World!", basic arithmetic, and user input/output 	2	2
2.	Variables, Expressions, and Control Flow: <ul style="list-style-type: none"> Variables, assignment, and naming conventions Arithmetic operations and precedence Conditional statements ('if', 'elif', 'else') Comparison operators (==, !=, <, >, etc.) Lab: <ul style="list-style-type: none"> Write programs to perform simple calculations based on user input Implement a simple number guessing game using 'if' conditions 	4	4
3.	Loops (For and While): <ul style="list-style-type: none"> Introduction to loops: 'for' and 'while' loops Using 'range()' with 'for' loops Infinite loops and loop control with 'break', 'continue', and 'else' Lab: <ul style="list-style-type: none"> Write programs using loops: summing numbers, factorial, counting even/odd numbers Practice with 'while' loops for creating interactive programs 	4	4



4.	Functions: <ul style="list-style-type: none"> Defining functions and calling functions Function parameters, return values, and scope (local vs global) Basic recursion Lab: <ul style="list-style-type: none"> Create simple functions for mathematical operations (e.g., area of a circle, power function) 	4	4
5.	Lists and List Operations: <ul style="list-style-type: none"> Introduction to lists (creation, indexing, and slicing) List methods: <code>append()</code>, <code>extend()</code>, <code>remove()</code>, <code>pop()</code>, <code>sort()</code> Iterating over lists with <code>for</code> loops List comprehensions Lab: <ul style="list-style-type: none"> Practice list operations (e.g., store and manipulate lists of numbers, and strings) Create a program that processes a list of student grades or temperatures 	4	4
6.	Working with Files: <ul style="list-style-type: none"> Reading and writing text files File handling modes (<code>'r'</code>, <code>'w'</code>, <code>'a'</code>) Working with file objects Handling file errors using <code>'try-except'</code> Lab: <ul style="list-style-type: none"> Write a program that reads input from a file and processes it Implement a program that logs output to a file (e.g., save user inputs) 	2	2
7.	Introduction to NumPy (Numerical Computing): <ul style="list-style-type: none"> Introduction to NumPy and arrays Creating arrays and array indexing Array operations: element-wise arithmetic, broadcasting Lab: <ul style="list-style-type: none"> Install NumPy and create basic arrays Perform operations on NumPy arrays (e.g., matrix addition, element-wise multiplication) 	4	2
8.	Introduction to Pandas (Data Manipulation): <ul style="list-style-type: none"> Introduction to Pandas and DataFrames Loading, exploring, and cleaning data with Pandas Indexing, slicing, and filtering data Basic data analysis with Pandas Lab: <ul style="list-style-type: none"> Load datasets using Pandas and perform data-cleaning tasks Write programs to analyze simple datasets (e.g., summary statistics, missing data handling) 	4	4
9.	Testing, Debugging, Exceptions, and Assertions: <ul style="list-style-type: none"> Input validation Exception handling. Error conditions handling. 	2	4



Lab:

- Identify and correct errors in Python programs.
- Implement debugging techniques to fix logical and syntactical issues.
- Utilize exception handling to manage errors and ensure robustness in programs.

30

30

D. Students Assessment Activities

No	Assessment Activities *	Assessment timing (in week no)	Percentage of Total Assessment Score
1.	Quizzes	4 - 15	10%
2.	Labs	4 - 15	20%
3.	Project	7 - 15	10%
4.	Midterm	8	20%
5.	Final Exam	Final Week	40%

*Assessment Activities (i.e., Written test, oral test, oral presentation, group project, essay, etc.).

E. Learning Resources and Facilities

1. References and Learning Resources

Essential References

- Introduction to Python Programming and Data Structures, 3rd edition Published by Pearson (July 25, 2022) © 2023 Y Daniel Liang
- Guttag, John. Introduction to Computation and Programming Using Python: With Application to Understanding Data Second Edition. [MIT Press, 2016. ISBN: 9780262529624.](#)
- Mark Lutz (2025). Learning Python, 6th Edition. Publisher(s): O'Reilly Media, Inc. ISBN: 9781098171308
- McKinney, W. (2022). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and Jupyter. 3rd Edition. O'Reilly Media.
- Grus, J. (2019). Data Science from Scratch: First Principles with Python. 2nd Edition. O'Reilly Media.

Supportive References

- Hill, C. (2021). Python for Beginners: Learn Python Programming from Scratch.
- Raschka, S., & Mirjalili, V. (2019). Python Machine Learning: Machine Learning and Deep Learning with Python, Scikit-Learn, and TensorFlow. 3rd Edition.





Electronic Materials	<ul style="list-style-type: none"> Umm Al Qura e-learning system containing teaching resources (Slides, assignment papers, etc.) Documentation: Python Software Foundation. Video Tutorials: Corey Schafer's Python Playlist. Interactive Tutorials: Real Python.
Other Learning Materials	N/A

2. Required Facilities and equipment

Items	Resources
facilities (Classrooms, laboratories, exhibition rooms, simulation rooms, etc.)	Lecture room with: * at least 40 seats * A data show projector connected to a PC preferably with an Internet connection * PC Lab (at least 20 seats)
Technology equipment (projector, smart board, software)	projector
Other equipment (depending on the nature of the specialty)	

F. Assessment of Course Quality

Assessment Areas/Issues	Assessor	Assessment Methods
Effectiveness of teaching	Students	Indirect Course survey and students' feedback.
Effectiveness of Students	Faculty Members, Peer Reviewers	Direct Report on the satisfaction of exam standards.
Quality of learning resources	Faculty Member, Course Coordinators	Direct Learning resources evaluation survey.
The extent to which CLOs have been achieved	Faculty Members, Program Leaders	Direct Course reports.





Assessment Areas/Issues	Assessor	Assessment Methods
Other		

Assessors (Students, Faculty, Program Leaders, Peer Reviewer, Others (specify))

Assessment Methods (Direct, Indirect)

G. Specification Approval

COUNCIL /COMMITTEE	Umm Al-Qura University Council
REFERENCE NO.	851281214463/193664
DATE	1447/01/20

