جـامعـة الـزيـتـونـــة الأردنيـة

Al-Zaytoonah University of Jordan

كلية العلوم وتكنولوجيا المعلومات

Faculty of Science and Information Technology

"عراقة وجودة "

"Tradition and Quality"

QF01/0408-4.0E	**Course Plan for Bachelor program - Study Plan Development and Updating Procedures/ Artificial Intelligence Department**

Study plan No.	**2021/2022**		**University Specialization**		**Artificial Intelligence**	
Course No.	**0142220**		**Course name**		**Data Science and Artificial Intelligence programming 1**	
Credit Hours	**3 hours**		**Prerequisite Co-requisite**		**Programming principles**	
Course type	□ **MANDATORY UNIVERSITY REQUIREMENT**	□ **UNIVERSITY ELECTIVE REQUIREMENTS**	□ **FACULTY MANDATORY REQUIREMENT**	□ **Support course family requirements**	☑ **Mandatory requirements**	□ **Elective requirements**
Teaching style	□ **Full online learning**		□ **Blended learning**		☑ **Traditional learning**	
Teaching model	□ **2 Synchronous: 1asynchronous**		□ **2 face to face : 1synchronous**		☑ **3 Traditional**	

Faculty member and study divisions' information (to be filled in each semester by the subject instructor)

Name	**Academic rank**	**Office No.**	**Phone No.**	**E-mail**	
To be filled by the instructor					
Division number	**Time**	**Place**	**Number of students**	**Teaching style**	**Approved model**
To be filled by the instructor					

Brief description

This course presents the python language to write scripts by applying basic and advanced concepts such as variables, conditional statements, strings, methods, lists, tuples dictionaries etc. Additionally, understanding the fundamentals of object-oriented programming in Python.

Learning resources

Course book information (Title, author, date of issue, publisher ... etc)	Paul Deitel and Harvey Deitel, "Intro to Python for Computer Science and Data Science: Learning to Program with AI, Big Data and the Cloud", Pearson Education, 2020.			
Supportive learning resources (Books, databases, periodicals, software, applications, others)	1. Dan Bader, Joanna Jablonski and Fletcher Heisler, "Python Basics: A Practical Introduction to Python 3", 4th ed, Ron Holland Designs, 2021. 2. John Hunt , "A Beginners Guide to Python 3 Programming", Springer International Publishing, Aug 13, 2019 3. John V. Guttag , "Introduction to Computation and Programming Using Python with Application to Understanding Data", MIT press,2017			
Supporting websites	https://docs.python.org/			
The physical environment for teaching	☑ Class room	□ labs	□ Virtual educational platform	□ Others
Necessary equipment and software	PyCharm : https://www.jetbrains.com/pycharm/ Or Anaconda: https://www.anaconda.com/			

QF01/0408-4.0E	**Course Plan for Bachelor program - Study Plan Development and Updating Procedures/ Artificial Intelligence Department**

Supporting people with special needs	-------------------
For technical support	-------------------

Course learning outcomes (S*= Skills, C= Competences K= Knowledge,)*

No.	**Course learning outcomes**	**The associated program learning output code**
	Knowledge	
K1	**Understand fundamentals of python such as variables, conditional statements and functions.**	**MK2**
K2	**Processing Strings and Sequences.**	**MK2**
K3	**Understand fundamentals of object-oriented programming in Python.**	**MK2**
	Skills	
S1	**Knowledge of the structure and model of the Python programming language.**	**MS2**
S2	**Use the Python programming language for various programming applications.**	**MS2**
S3	**Develop software in the Python programming language using OOP and comparing it with OOP in Java.**	**MS2**
	Competences	
C1	**The ability to write basic python scripts.**	**MC1**
C2	**The ability to process strings.**	**MC1**
C3	**The ability to write scripts using lists, tuples and Dictionaries.**	**MC1**
C4	**The ability to apply OOP concepts in Python.**	**MC1**

Mechanisms for direct evaluation of learning outcomes

Type of assessment / learning style	**Fully electronic learning**	**Blended learning**	**Traditional Learning (Theory Learning)**	**Traditional Learning (Practical Learning)**
First exam	0	0	%20	**0**
Second / midterm exam	%30	%30	%20	**30%**
Participation / practical applications	0	0	10	**30%**
Asynchronous interactive activities	%30	%30	0	**0**
final exam	%40	%40	%50	**40%**

Note: Asynchronous interactive activities are activities, tasks, projects, assignments, research, studies, projects, and work within student groups ... etc, which the student carries out on his own, through the virtual platform without a direct encounter with the subject teacher.

Schedule of simultaneous / face-to-face encounters and their topics

Week	**Subject**	**learning style***	**Reference ****
1	**Introduction to Python Programming** Variables and Assignment Statements Arithmetic Function print and an Intro to Single- and Double-Quoted Strings Triple-Quoted Strings Getting Input from the User	Lectures	49-59
2	**Control Statements and Program Development** Comparison Operations Logical Operators	Lectures	73-119

	Selection Control Statements		
3	If statements applications For loop and applications Else with loops	Lectures	73-119
4	While loops nested loops	Lectures	73-119
5	**Strings: A deeper look** String concatenation in operator string indexing and slicing	Lectures	238-250
6	string built in functions Applications		238-250
7	**Functions** Functions in python (math functions) User defined functions Variables Scope	Lectures	119-150
8	Keywords arguments Default arguments Random function Lambda function **Midterm Exam**	Lectures	119-150
9	**Sequences: Lists and Tuples** Declaring lists Filling lists by using append Some lists functions and methods	Lectures	155-199
10	List comprehensions Processing 2d lists Passing lists to functions	Lectures	155-199
11	Tuples Applications	Lectures	155-199
12	**Dictionaries and Sets** Creating a Dictionary Iterating through a Dictionary Basic Dictionary Operations	Lectures	209-215
13	Dictionary Methods keys and values Applications on dictionaries	Lectures	209-215
14	**Object-Oriented Programming** Defining classes in Python Instantiating an object Access Modifiers in python	Lectures	355-375
15	Inheritance Applications	Lectures	355-375
16	**Final Exam**		

*** Learning styles: Lecture, flipped learning, learning through projects, learning through problem solving, participatory learning ... etc.**

**** Reference: Pages in a book, database, recorded lecture, content on the e-learning platform, video, website ... etc.**