

QFO-AP-VA-008	رمز النموذج :	اسم النموذج : خطة المادة الدراسية	 جامعة فيلادلفيا Philadelphia University
2	رقم الإصدار: (Rev)	الجهة المصدرة: نائب الرئيس للشؤون الأكاديمية	
2021-5-4	تاريخ الإصدار:	الجهة المدققة : اللجنة العليا لضمان الجودة	
4	عدد صفحات النموذج :		

Course Title: Programming Fundamentals (2)		Course code: 750114	
Course Level: 1		Course prerequisite (s) and/or corequisite(s): 750113	
Lecture Time:		Credit hours: 3	
UR	☐	FR	☒
DR	☐	C	☐
E	☐		

Academic Staff Specifics

Name	Rank	Office Number and Location	Office Hours	E-mail Address
Dr. Mohammad Taye	Assistant professor	7309	ST: 10:30– 11:00 MW:11:00–12:30	mtaye@philadelphia.edu.jo

The Learning Style Used in Teaching the Course

<u>The Learning Style</u>			
Blended Learning ☐			
Electronic Learning ☐			
Face-to-Face Learning ☒			
Face-to-Face	Electronic	Blended	Percentage
100%			

Course module description

This course presents the fundamental concepts of programming using Python. It covers the basic structures of the programming tools such as variable names; data types; control structures; arrays; functions; Sequences (Strings, Tuples, Lists); Iteration; Dictionaries ; Set; Modules; Exceptions; introduction to file processing; and Introduction to Object oriented

Course module objectives

The objectives of this course are to:

1. Clarify the basic programming concepts including data types, variables, modularity, parameters, conditional statements, iteration, and arrays.
2. show and solve several core data structures: Lists, Dictionaries, Tuples, and Strings.
3. use and employ objects, functions and modularity.
4. Demonstrate methods of error handling
5. show input/output with files in Python.
6. outline basic and some advanced issues related to writing classes and methods

Course/ module components

- Books (title , author (s), publisher, year of publication)

Romano, Fabrizio (Author)

Publication Data: Birmingham: Mumbai Packt Publishing , 2018

ISBN: 978-1-78899-666-2

Learn Python programming: the no-nonsense, beginner's guide to programming, data science, and web development with Python 3.7

Teaching methods:

Duration: 16 weeks, 80 hours in total

Lectures: 32 hours (2 hours per week),

Tutorials: 16 hours (1 per week),

Laboratories: 32 hours, 2 per week

Student Learning Outcomes (SLO)

Learning outcomes achievement

A- Knowledge and understanding

A1. Identify a wide range of principles and tools available to the software developer, such as design methodologies, choice of algorithm, language, software libraries and user interface technique

A2. Clarify the professional and ethical responsibilities of the practicing computer professional including understanding the need for quality, security, and computer ethics.

B- Intellectual skills (thinking and analysis).

B1. Analyze a wide range of problems and provide solutions through suitable algorithms, structures, diagrams, and other appropriate methods

B2. Practice self-learning by using the e-courses

C- Practical skills

C1. assess the balance between self-reliance and seeking help when necessary in new situations

C2. Display personal responsibility by working to multiple deadlines in complex activities

D- Transferable Skills

D1. Prepare and deliver coherent and structured verbal and written technical reports.

D2. Design, write, and debug computer programs in appropriate languages

- Development: A1, A2 are developed through the lectures and laboratory sessions.

C2, D1, D2 are developed through Tutorials and Lab sessions,

B2, D1, D2 and C2 are developed through Homework

- Assessment : A2, B1, D1, and D2 and are assessed through Quizzes, written exams, and Practical Works Exams.

B2, D1, D2 and C2 are assessed through Homework Exam.

Assessment instruments:

Evaluation of students' performance (final grade) will be based on the following three categories:

<u>Allocation of Marks</u>	
Assessment Instruments	Mark
Mid examination	30%

Final examination	40%
Lab works, Quizzes, and tutorial contributions	30%
Total	100%

Documentation and academic honesty

- Protection by copyright; Avoiding plagiarism.

Course/module academic calendar

Course Academic Calendar		
Week	Subject	Lab works and tutorials
1	Introduction to Python Supporting Tools: Git (GitHub, bitbucket, etc.), Environment ()	Lab work #1 (Get started with python language environment program editing, compiling, executing, debugging) Tutorial 1
2	Basic syntax: basic data types; variables, assignments; immutable variables; numerical types; casting, arithmetic operators and expressions; comments in the program; conditionals and Boolean expressions	Lab work #2 Tutorial 2
3	Functions: definition and use, arguments, scope	Lab work #3 Tutorial 3
4	Functions: Lambda, recursion	Lab work #4 Tutorial 4
5	Sequences: Strings, Tuples, Lists Strings: indexing, slicing, modifying, concatenating, formatting, string methods	Lab work #5 Tutorial 5
6	Sequences: Strings, Tuples, Lists Lists: list literals, adding and removing items, accessing and replacing values, looping lists, list comprehension, copying lists, joining lists, searching and sorting lists, list methods Tuples: tuple literals, accessing and replacing values, unpack tuples, looping tuples, joining tuples, tuple methods	Lab work #6 Tutorial 6
7	Iteration, looping and control flow, ranges First Exam	Lab work #7 Tutorial 7
8	Dictionaries and Set	Lab work #8

	Dictionaries: dictionaries literals, adding and removing items, accessing and replacing values; looping dictionaries, copying dictionaries, dictionary methods.	Tutorial 8
9	Dictionaries and Set Set: sets literals, adding and removing items, accessing and replacing values, looping sets, joining sets, set methods	Lab work #9 Tutorial 9
10	Modules, import, and pip Modules: creating and using modules, variables in modules import: importing and renaming modules pip: listing installed packages, downloading, using, finding, and removing packages	Lab work #10 Tutorial 10
11	Exceptions and Testing Exceptions: Exception handling, multiple exceptions, else, finally, and raise an exception Testing: Manual testing, understanding error messages 2 nd Exam	Lab work #11 Tutorial 11
12	Files, Text Processing, CSV, and JSON Files and Text Processing: OS and sys modules, reading/writing text and numbers from/to a file	Lab work #12 Tutorial 12
13	Files, Text Processing, CSV, and JSON CSV: Creating and reading a CSV formatted file JSON: Convert from JSON to Python and vice versa	Lab work #13 Tutorial 13
14	Introduction to Object oriented	Lab work #14 Tutorial 14
15	Introduction to Object oriented	Lab work #15 Tutorial 15
16	Review Final Exam	

[[Expected workload

On average students need to spend 2 hours of study and preparation for each 50-minute lecture/tutorial.

Attendance policy

Absence from lectures and/or tutorials shall not exceed 15%. Students who exceed the 15% limit without a medical or emergency excuse acceptable to and approved by the Dean of the relevant college/faculty shall not be allowed to take the final examination and will receive a fail grade for the course. If the excuse is approved by the Dean, the student shall be considered to have withdrawn from the course.

Module references

Books:

Students will be expected to give the same attention to these references as given to the Module textbook(s)

1. Gutttag, John V. (Author), "Introduction to computation and programming using python", New Delhi: PHI Learning Private Limited, 2014
2. Mark Lutz and David Ascher, "Learning Python", Beijing: O'Reilly, 2004, 2nd ed..
3. Deitel, H. M, " Python : how to program ", Upper Saddle River, New Jersey: Prentice Hall, 2002
4. Dusty Phillip , "Python 3 Object Oriented Programming", Packt, July 2010. (ISBN : 1849511268 ISBN 13 : 978-1-849511-26-1)
5. Allen B. Downey, " Python for Software Design How to Think Like a Computer Scientist", Olin College of Engineering, Massachusetts, May 2009. (ISBN-13: 9780521898119)

Website

www.python.org

<https://www.w3schools.com/python/>

<https://www.tutorialspoint.com/python/index.htm>

<https://www.learnpython.org/>

<https://www.guru99.com/python-tutorials.html>

<https://www.programiz.com/python-programming>

<https://realpython.com/>