|  |  |  |  |
| --- | --- | --- | --- |
| 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 |
| 4-5-2021 | **تاريخ الإصدار:** | **الجهة المدققة :** اللجنة العليا لضمان الجودة |
| 4 | **عدد صفحات النموذج :** |

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

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

**The Learning Style Used in Teaching the Course**

|  |  |  |  |
| --- | --- | --- | --- |
| **The Learning Style** | | | |
| **Blended Learning** | | | |
| **Electronic Learning** | | | |
| **Face-to-Face Learning X** | | | |
| **Percentage** | **Blended** | **Electronic** | **Face-to-Face** |
|  |  | **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. Demonstrate an understanding of basic programming concepts including data types, variables, modularity, parameters, conditional statements, iteration, and arrays.

2. Understand and manipulate several core data structures: Lists, Dictionaries, Tuples, and Strings.

3. Understand and employ objects, functions and modularity.

4. Demonstrate methods of error handling

5. To do input/output with files in Python.

6. Demonstrate an understanding of basic and some advanced issues related to writing classes and methods

**Course/ module components**

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

[Romano](http://library.philadelphia.edu.jo/scripts/minisa.dll/54/PAUTHOR/Romano?KEYSEARCH&DISPLAY=AUTHORS+), 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)**

At the end of the course, students should be able to

Knowledge and Understanding

A2: Understand a wide range of principles and tools available to software developers, including design methodologies, choice of algorithms, programming languages, software libraries, and user interface techniques.

A4: Know and understand the software and hardware used in the development of computer systems.

Intellectual Skills

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

B4: Practice self-learning by utilizing e-courses and online resources.

Professional and Practical Skills

C3: Work effectively with others in a collaborative environment.

C4: Strike a balance between self-reliance and seeking help when necessary in new situations.

General and Transferable Skills

D2: Prepare and deliver coherent and structured verbal and written technical reports.

D4: Use scientific literature effectively and make discerning use of web resources.

Mapping Table Between PLOs and CLOs

### Mapping Table Between PLOs and CLOs

| **PLOs** | **CLOs** |
| --- | --- |
| **PLO 1:** Demonstrate knowledge of fundamental programming concepts. | **CLO A2:** Understand a wide range of principles and tools available to software developers. |
| **PLO 2:** Apply problem-solving techniques using appropriate algorithms and data structures. | **CLO B1:** Analyze a wide range of problems and provide solutions through suitable algorithms. |
| **PLO 3:** Collaborate effectively in team environments to develop software solutions. | **CLO C3:** Work effectively with others in a collaborative environment. |
| **PLO 4:** Develop written and oral communication skills for technical reporting. | **CLO D2:** Prepare and deliver coherent and structured verbal and written technical reports. |
| **PLO 5:** Demonstrate ethical practices in software development and research. | **CLO A4:** Know and understand the professional and ethical responsibilities of the practicing computer professional. |

**Learning outcomes achievement**

* Development: A2, A4, and A5 are developed through the lectures and laboratory sessions.

B1, D5, C3, and C4 are developed trough Tutorials and Lab sessions,

B4, D2, D4, D5, and C5 are developed through Homework

* Assessment : A2, A4, A5, B1, D5, and C4 and are assessed through Quizzes, written exams, and Practical Works Exams.

B4, D2, D4, D5, and C5 are assessed through Homework Exam.

**Assessment instruments:**

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

|  |  |
| --- | --- |
| **Allocation of Marks** | |
| **Mark** | Assessment Instruments |
| **30%** | Mid examination |
| **40%** | Final examination |
| **30%** | Lab works, Quizzes, and tutorial contributions |
| **100%** | Total |

**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**  **Dictionaries**: dictionaries literals, adding and removing items, accessing and replacing values; looping dictionaries, copying dictionaries, dictionary methods. | **Lab work #8**  **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  2nd 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 well 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. Guttag, 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](http://www.cambridge.org/catalogue/searchResult.asp?ipcode=242501&sort=Y), " 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](http://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/>