AMERICAN UNIVERSITY OF BEIRUT
MAROUN SEMAAN FACULTY OF ENGINEERING & ARCHITECTURE

INTRODUCTION TO COMPUTATION AND PROGRAMMING SYLLABUS

Online Certificate

CERTIFICATE DESCRIPTION

This certificate is an introduction to programming with an emphasis on abstractions and elementary algorithmic ideas. It uses the Python programming language. Topics include data types, selection, repetition, lists, tuples, strings, functions, files, plotting, exception handling, program efficiency, recursion, divide and conquer algorithms, recurrence relations, sorting and searching algorithms, binary search, merge sort, randomized quicksort, dictionaries, memoization, classes and object oriented programming, stacks and queues, applications, and selected topics.

Introduction to Computation and Programming Using Python

Delivery Methods

The program will be delivered in an asynchronous mode

Certificate Instructor

Prof. Louay Bazzi,
ECE department, MSFEA, AUB

Required materials

Guttag, John. Introduction to Computation and Programming Using Python: With Application to Understanding Data, Second Edition. MIT Press, 2016

HOW IS IT STRUCTURED?

- The program is divided into three modules, with each module further divided into topics. Each topic consists of multiple lessons, and each lesson is accompanied by prerecorded interactive short videos. Additionally, each topic includes programming assignments to reinforce learning.

ASYNCHRONOUS MATERIAL

- The learning material consists of concise videos, typically lasting around 5 minutes each, complemented by supporting resources. The decision to keep the videos short is based on research indicating that students often struggle to maintain focus during lengthy video presentations. These brief videos serve as the primary asynchronous content. In addition, the supporting asynchronous material includes slide sets corresponding to each topic, combined videos aligned with specific lessons, code samples, and handouts.

LEARNING BY DISCOVERING

- Each short video concludes with a question, and the answer to that question is provided in the subsequent video. It is highly recommended to allocate ample time to ponder the question before proceeding to the next video. This approach aims to enhance your problem-solving skills by promoting active learning and shifting you from being a passive recipient of instruction to an engaged learner. Some questions may not have straightforward answers, and it is normal for it to take some time to arrive at a complete solution.

  It is enough to contemplate the question for a period that suits your schedule, allowing your mind to prepare for the answer in the subsequent video. The ultimate goal is not solely to find the correct answer but to recognize that identifying what doesn't work is equally valuable in the learning process.

One of the primary goals of this course is to foster a computer scientist's mindset, which involves appreciating the significance of understanding what doesn't work in order to appreciate what does work.

The questions presented in the course have been gathered from previous iterations of the course that followed a traditional delivery format. In those instances, students were provided with only a few minutes to propose solutions and suggestions. However, with the asynchronous format of this course, you now have the advantage of allocating as much time as necessary, be it a few minutes or even hours, depending on your individual time

PROGRAMMING ASSIGNMENTS

- Programming assignments play a vital role in this program, allowing you to develop both coding and problem-solving skills. The assignments encompass problems of diverse difficulty levels, some of which include hints. In certain cases, hints are deliberately presented in small and backward font to encourage you to contemplate the problem before resorting to them. On average, completing an assignment typically takes around 3 to 5 hours.

  You are responsible for working on the assignments both during your lab sessions and at home. It is crucial that you dedicate your own effort to completing the assignments in order to achieve the intended learning outcomes of the program. While discussing your assignments with classmates and teaching assistants can be beneficial, it is expected that you submit your own individual work. The solutions to the assignments will be made available immediately after the due dates. Please bear in mind that there is a significant distinction between attempting to solve a problem or actively working on it, and simply reading the solutions once they are released. To develop your problem-solving skills effectively, it is expected that you first tackle the assignment on your own and then compare your work with the posted solutions.

  Assignments will be evaluated using an automatic online judge, allowing you to submit each problem multiple times if needed. If your submission is not entirely correct, the online judge will provide feedback regarding the specific tests that did not pass. Once you have completed the debugging process on the online judge, you are expected to submit your final assignment on Moodle, the online learning platform. Upon submission on Moodle, you will gain access to the solutions for the assignment.

FINAL EXAM

- In order to get a certificate of completion, it is necessary to participate in the final exam, which will be conducted online. The specific date of the final exam will be announced on Moodle. To successfully pass the final exam, it is expected that you have completed the program's asynchronous material and the assignments.

TOPICS

UNIT I Foundations

Topic	Description	Week
1,2,3	Introduction to computation using Python, data types, selection, repetitions, and bisection method	1 & 2
4,5	Lists, tuples, strings, and functions	3 & 4
6,7	Files, exception handling, plotting, and Monte Carlo simulation	5
8	Introduction to program efficiency and asymptotic analysis, binary search, and insertion sort	6

UNIT II Recursion, memoization, searching and sorting, divide and conquer, recurrence relations, data structures, and applications

Topic	Description	Week
9,10	Recursion: elementary examples, memoization, merge sort, divide and conquer algorithms, recurrence relations, recursion tree method	7 & 8
11	Elementary data structures: two-dimensional lists, dictionaries, and stacks	9
12	Applications: randomized quick sort, recursive enumeration, maze depth-first traversal	10

UNIT III Object Oriented Programming with applications and selected topics

Topic	Description	Week
13	Object Oriented Programming, classes, and inheritance	11 & 12
14	Implementation of stacks and queues	12
15	Graphs: representation, depth-first search, and breadth-first search	13

LEARNING OUTCOMES

At the end of the program, students will be able to:

- Apply the principles of functional programming
- Implement searching and sorting algorithms
- Solve computational problems using searching and sorting
- Analyze the efficiency of elementary algorithms
- Solve computational problems using recursion
- Solve computational problems using elementary data structures such as two-dimensional lists, dictionaries, stacks, and queues
- Apply the principles of object-oriented programming

COLLABORATION POLICY AND CHEATING

Students are expected to complete all work with the highest standard of integrity in line with AUB's Student Code of Conduct. Plagiarism, forgery, cheating or any form of academic misconduct will not be tolerated and will automatically result in a failing grade. In this program, cheating is defined as follows:

- Copying or partially copying an assignment code from any person or source as is or with minor changes such as:
  - Editing the name and Id in the beginning of the file
  - Rephrasing/adding comments in the code
  - Changing variable names
  - Reordering some instructions
  - Rewriting.
- Getting help without acknowledgement for the sake of merely finishing the assignment.
- Sharing your code with anyone who might copy your code and submit it under their name.
- Providing or receiving step-by-step coaching on solving the assignment problems without ensuring that the learner fully understands what is going on.

SIGN UP

Register now to enroll in the interactive program!

CLICK TO REGISTER

Registration Link

fswd@aub.edu.lb