Course Syllabus

1. Course number and name: 020IF3NI4 Programming 3

2. Credits and contact hours: 4 credits, 2x1:15 course hours

3. Instructor's or course coordinator's name: Charbel AOUAD

4. Text book, title, author, and year
   a. other supplemental materials:
      Professor textbook and course material

5. Specific course information
   a. catalog description :
      This course covers advanced programming concepts using Python. Topics include a systematic study of existing sorting algorithms and how to calculate their time complexity. Topics also include file management to save or read structured or unstructured data, writing user-friendly command-line interfaces, connecting to remote hosts/services in order to retrieve or submit data via Application Programming Interfaces.

   b. prerequisites : 020IF1NI2 Programming 1

   c. Required/Elective/Selected Elective: Required

6. Specific goals for the course
   a. specific outcomes of instruction
      - Explain and write the code for the basic sorting algorithms
      - Analyze asymptotic behavior of sorting algorithms
      - Write the code for recursive sorting algorithms using recursive functions
      - Determine and apply the timeliest efficient sorting algorithm on the given data
      - Use Python file management to save and read data from text or binary files
      - Write command-line interfaces
      - Use an application programming interface to manipulate data of remote systems

   b. KPIs addressed by the course.

KPI	a1	c2	g1		
Covered	x	x	x		
Assessed	x	x	x		
Give Feedback	x	x	x		

7. Topics and approximate lecture hours :

- Introduction to sorting algorithms (1 Lecture)
- Time complexity and asymptotic analysis of an algorithm (2 Lectures)
- Bubble sort : algorithm and time complexity (2 Lectures)
- Selection sort : algorithm and time complexity (2 Lectures)
- Simple and binary insertion sort : algorithms and time complexity (2 Lectures)
- Recursive algorithms : concept and examples (2 Lectures)
- Merge sort : algorithm and time complexity (2 Lectures)
- Quick sort : algorithm and time complexity (2 Lectures)
- Text and binary files : save and read structured and unstructured data (4 Lectures)
- Command line interface for Python tools/programs (2 Lectures)
- Remote data access (2 Lectures)
- Usage of an external Application Programming Interface (2 Lectures)
- Lab sessions (numpy/scilab/matplot lib,...) (3 Lectures)