


QFO-AP-FI-MO02	اسم النموذج: Course Syllabus	جامعة فيلادلفيا
رقم الاصدار : 1 (Revision)	الجهة المصدرة: كلية تكنولوجيا المعلومات	
التاريخ: 2017/11/05	الجهة المدققة: عمادة التطوير والجودة	Philadelphia University
عدد صفحات النموذج:		

<u>Course Syllabus</u>	
<b>Course Title: Introduction to Software Engineering</b>	<b>Course code: 0721110</b>
<b>Course Level: 1</b>	<b>Course prerequisite (s): 0750113 + 0731110</b>
<b>Lecture Time: 11:10- 12:00</b>	<b>Credit hours: 3</b>

<u>Academic Staff Specifics</u>				
Name	Rank	Office Number and Location	Office Hours	E-mail Address
Enas Naffar	Lecturer	7333		enaffar@philadelphia.edu.jo

#### **Course module description:**

This course is a first introductory course to the Software Engineering discipline. It covers, in a concise way, the following major topics of Software engineering: Software Process, Software Requirements, Software Architecture, Software Design and Software Testing. This course will focus on theoretical and practical aspects of the discipline.

#### **Course module objectives:**

The objective of this course is twofold:

- Teach students the skills needed to execute and smallish commercial project.
- Provide students with the necessary conceptual background for undertaken advanced studies in software engineering, through specialized software engineering courses.

## Course/ module components

- **Books (title , author (s), publisher, year of publication)**
  1. Title: A Concise Introduction to Software Engineering  
Author(s)/Editor(s): Pankaj Jalote  
Publisher: Springer, 2008
- **Support material (s): slides, textbook**

### Teaching methods:

Duration: 15 weeks, 45 hours in total

Lectures: 36 hours, Tutorial: 5 hours, Exams: 4 hours

### Learning outcomes:

A- Knowledge and understanding.

- 1) Be aware of software development features (quality, schedule, cost). (A1)
- 2) Recognize the benefits of applying a systematic approach to develop software.(A1)
- 3) Recognize the different stages of software development.(A1)
- 4) Understand the basic concepts related to requirement, design, coding, testing and maintenance phases of a software development process.(A1)
- 5) Recognize different software processes approaches. (A1)

B- Cognitive skills (comprehension and analysis).

- 1) Distinguish between structural analysis and design approach and object oriented analysis and design approach. (B5)
- 2) Conduct the different stages of a software development from requirements, to analysis, to design, to coding and testing. (B2, B3, B5)
- 3) Identify a range of solutions and critically evaluate and justify proposed design solutions. (B4)

C- Practical and subject specific skills.

- 1) Write a technical documentation of a project. (C7)
- 2) Conduct a dialogue with software stakeholders (requirements gathering). (C9)

D- Transferable Skills.

- 1) Effectively participate in team-based activities. (D6)
- 2) Structure and communicate ideas effectively, both orally, in writing, and in cases involving a quantitative dimension. (D4)
- 3) Use IT skills and display mature computer literacy. (D8)
- 4) Work independently and with others.(D6)

### Assessment of Learning Outcomes

Learning outcomes of A and B are assessed by examinations and tutorials. Learning outcomes of C and D are assessed by assignments and laboratory.

<u>Allocation of Marks</u>	
<b>Assessment Instruments</b>	<b>Mark</b>
First examination	<b>20 marks</b>
Second examination	<b>20 marks</b>
Final examination:	<b>40 marks</b>
Quizzes, Home works	<b>20 marks</b>
Total	<b>100 marks</b>

### Documentation and academic honesty

- Documentation style (with illustrative examples)
- Protection by copyright
- Avoiding plagiarism.

### Course/module academic calendar

<b>week</b>	<b>Basic and support material to be covered</b>	<b>Homework/reports and their due dates</b>
(1)	<b>The Software Problem: 1.1</b>	
(2)	<b>The Software Problem: 1.2; 1.3</b>	
(3)	<b>Software Processes: 2.1; 2.2; 2.3.1</b>	
(4)	<b>Software Processes: 2.3.2; 2.3.3; 2.3.7</b> <b>Tutorial 1</b>	
(5)	<b>Software Requirements Analysis and Specification: 3.1; 3.2; 3.3</b>	
(6) <b>First examination</b>	<b>Software Requirements Analysis and Specification: 3.5.1; 3.6</b>	
(7)	<b>Software Requirements Analysis and Specification: 3.6</b> <b>Tutorial 2</b>	<b>First Homework</b>
(8)	<b>Software Architecture: 5.1; 5.2;</b>	
(9)	<b>Software Architecture: 5.3; 5.4</b>	
(10)	<b>Software Architecture: 5.6</b> <b>Tutorial 3</b>	
(11)	<b>Software Design: 6.1</b>	

<b>Second examination</b>		
(12)	<b>Software Design: 6.2 Tutorial 4</b>	<b>Second Homework</b>
(13)	<b>Software Design: 6.4.1; 6.5 Tutorial 5</b>	
(14)	<b>Testing</b>	
(15)	<b>Revision &amp; Seminars</b>	
(16) <b>Final Examination</b>		

**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 shall receive a mark of zero 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:**

- a. Software Engineering: A practitioner's Approach, R.S. Pressman, Mc Graw Hill, 2010.
- b. Software Engineering: Theory and Practice, S. L. Pfleeger & J. M. Atlee, Pearson Prentice Hall, 2006
- c. Object Oriented and Classical Software Engineering, S. R. Schach, Mc Graw Hill. 2005.