


QFO-AP-FI-MO02	اسم النموذج: Course Syllabus	جامعة فيلادلفيا
رقم الإصدار: 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>	
Course Title: Software Architecture	Course code: 721320
Course Level: 3	Course prerequisite (s) and/or co-requisite (s): Software Modeling(721222)+Software Requirements(721230)
Lecture Time:13:10-14:00	Credit hours: 3

<u>Academic Staff</u>				
<u>Specifics</u>				
Name	Rank	Office Number and Location	Office Hours	E-mail Address
Dr. Moayad A. Fadhil	Ass. Prof.	7305	10:00-13:00	mathami@philadelphia.edu.jo

### Course module description:

Successful design of complex software systems requires the ability to describe, evaluate, and create systems at an architectural level of abstraction. This course introduces architectural design of complex software systems. The course considers commonly-used software system structures, techniques for designing and implementing these structures, models and formal notations for characterizing and reasoning about architectures, tools for generating specific instances of an architecture, and case studies of actual system architectures. It teaches the skills and background you need to evaluate the architectures of existing systems and to design new systems in principled ways using well-founded architectural paradigms.

### Course module objectives:

This course aims to:

1. Show and emphasize the architecture's critical role in software engineering
2. Introduce the basic concepts of software architecture
3. Teach a set of commonly used architecture styles and architecture patterns.
4. Present techniques and approaches for designing applications from an architecture-centric perspective.

## Course/ module components

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

1. Title: Software Architecture: Foundations, Theory, and Practice  
Author(s)/Editor(s): R. N. Taylor, N. Medvidovic, and E. M. Dashofy  
Publisher: John Wiley & Sons, 2010.  
ISBN-10: 0470167742  
ISBN-13: 978-0470167748

2. Title: Software Architecture in practice  
Author(s)/Editor(s): Len Bass, Paul Clements and Rick Kazman  
Publisher: Addison-Wesley, 2007

- **Support material:** Textbook lecture slides

## Teaching methods:

Duration: 15 weeks, 45 hours in total  
Lectures: 35 hours, Tutorial: 8 hours  
Exams (first and second): 2 hours

## Learning outcomes:

- Knowledge and understanding:

1. Understand the influence of architectural drivers on software structures.(A1)
2. Understand the technical, organizational, and business role of software architecture.(A4)
3. Understand and distinguish the concepts of Software Components and Connectors.(A2)
4. Understand the concept of Architecture Styles and patterns(A3)
5. Understand the principles of good architectural documentation.(A1)

- Intellectual skills (thinking).

1. Specify, model, and design architectural patterns/styles alternatives for a problem and choose among them.(B1, B3)
2. Analyze and Evaluate the fitness of an architectural design in meeting a set of system requirements and balancing quality trade-offs.(B6)

- Practical skills.

1. Use software architecture technologies (CASE tools) for each phase of architecture development.(C1, C2)
1. Prepare a good software architectural documentation.(C7)
2. Interact with the stakeholders in order to develop a good software architecture(C9)

- 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.(D5)

- 3) Use IT skills and display mature computer literacy.(D8)
- 4) Work independently and with others.(D4)

**Assessment of Learning Outcomes**

Learning outcomes (A1, A2, A3, A4) are assessed by examinations; Learning outcomes (B1,B3, B6, C7, C8, D8) are assessed by examinations, tutorials, projects/assignments; Learning outcomes (C1, C2, C7, C9, D4, D5, D6) are assessed by laboratory works and projects.

**Assessment instruments**

- Two written exams
- Project
- Three Course works
- Final (written) examination: 40 marks

<b><u>Allocation of Marks</u></b>	
<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>
Coursework , Project	<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)	Architecture Introduction: The Big Idea	
(2)	Architecture in Context: The Reorientation of the Software Engineering	
(3)	Basics Concepts of Software Architecture(Styles)	
(4)	Basics Concepts of Software Architecture(Patterns)	<b>First Coursework</b>
(5)	Designing Architectures(Design Process)	
<b>(6) First examination</b>	Designing Architectures(Architectural Conception)	
(7)	Architecture Styles (Traditional Language and Layers)	
(8)	Architecture Styles (Data flow and Shard memory)	<b>Second Coursework</b>
(9)	Architecture Styles (Interpreter and Implicit Invocation)	
(10)	Architecture Styles (Peer to peer and Complex Styles)	
<b>(11) Second examination</b>	Connectors (Roles)	
(12)	Connectors (Types)	
(13)	Connectors (Chosen Connectors )	
(14)	Modeling(Concepts)	<b>Third Coursework</b>
(15)	Modeling(Evaluating)	
<b>(16) Final Examination</b>	<b>Project Presentation</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**

Title: The Unified Software Development Process  
Author(s)/Editor(s): Jacobson, Booch and Rumbaugh  
Publisher: Addison Wesley, 2003

Title: Object-Oriented Software Engineering: Using UML, Patterns, and Java  
Author(s)/Editor(s): Bernd Bruegge, Allen Dutoit  
Publisher: Prentice Hall, 2003

Title: Modern Systems Analysis and Design  
Author(s)/Editor(s): J. Hoffer, and J. Valacich,  
Publisher: Prentice Hall

Title: Software engineering 7/e  
Author(s)/Editor(s): I. Sommerville  
Publisher: Addison Wesley, 2004  
Publisher: Addison Wesley, 2002

Title: Object-Oriented Analysis and Design  
Author(s)/Editor(s): Grady Booch  
Publisher: Addison Wesley, 2002

#### **Website:**

[www.isr.uci.edu/projects/archstudio/](http://www.isr.uci.edu/projects/archstudio/).