

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>	
Course Title: System Analysis and Design	Course code: 0721322
Course Level: 3	Course prerequisite: Software Architecture(721320)
Lecture Time:09:10-10:00	Credit hours: 3

<u>Academic Staff 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:

This course completes the student knowledge on Software Design. This course introduces the major design goals (correctness, reusability, robustness, flexibility). Then the course focuses on basic concepts of software architecture, component technologies, architectural design principles and design patterns.

Course module objectives:

The objective of this course is to introduce and detail the factors and the practices that tend to produce good quality software designs.

Course/ module components

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

1. Software Design: from programming to architecture, Braude Eric, John Wiley & sons, 2004.
2. The Art of Software Architecture: Design Methods and Techniques
Stephen T. Albin , John Wiley & Sons, 2003
3. Introduction to Software Engineering Design: Processes, Principles, and Pattern with UML2,
Christopher Fox, Addison-Wesley Computing, 2006.

Support material: Textbook slides

Teaching methods:

Lectures, tutorials, problem solving, programming practices

Learning outcomes:

Knowledge and understanding:

1. Explain the need for describing software systems with models, as a way to abstract from the system’s complexity and to reason about its properties. (A2)
2. Explain the role, the importance and benefits of design in the different phases of software development. (A1)
3. Recall and explain the principles used to deal with system complexity (abstraction, refinement, decomposition, hierarchy, separation of concerns). (A2)
4. Recall and explain the different design perspective/views (Structure, OOD) (A2)
5. Know the various UML diagrams: object diagram, class diagram, state diagram, sequence diagrams, activity diagrams. (A2)
6. Explain the various concepts related to a subset of UML diagrams (object, class, attribute, event, action, activity, transition, association, aggregation/composition, inheritance...). (A2)

Cognitive skills (thinking and analysis)

1. Build and design software models. (B1)
2. Apply abstraction and separation of concerns principles to build software model. (B5)
3. Analyze, transform, improve, validate models. (B2)
4. Evaluate software models. (B4)

Practical skills

1. Draw software Models using appropriate software modeling tools (C2)
2. Practice specific modeling languages (C3).
3. Prepare and deliver coherent and structured verbal and written technical report.(C7)
4. Use the scientific literature effectively. (C8)

Transferable skills

1. Solve problems (D3)
2. Use creativity (D2)
3. Communicate effectively with non-specialist as well as computer scientist, (D4)

Assessment of Learning Outcomes

Learning outcomes (A1-A2) are assessed by examinations; Learning outcomes (B1,B5, B2, B4, C7, C8, D3, D2, D4) are assessed by examinations, tutorials, projects/assignments; Learning outcomes (C2, C3) are assessed by laboratory works and projects.

Assessment instruments

- **Class works:** 15 (quizzes)
- **Practice** (case tool): 05
- **Final examination:** 40
- **Short Examinations:** 2 x 20

Assessment instruments

- Quizzes.
- Home works: two programming projects
- Final examination: 40 marks

Allocation of Marks	
Assessment Instruments	Mark
First examination	20
Second examination	20
Final examination	40
Quizzes, Projects	20
Total	100

Documentation and academic honesty

- Protection by copyright
- Avoiding plagiarism.

week	Basic and support material to be covered	Homework/reports and their due dates
(1)	Software Design: purpose, motivation, design levels Software Design Principles (I): Correctness, Robustness	
(2)	Software Design Principles (II): Flexibility, Reusability, Efficiency	
(3)	Software Requirements	
(4)	- Object-oriented analysis (Unified Process)(I) Building Analysis Model : Identifying the candidate classes and objects Define the interaction between objects	First Homework
(5)	- Object-oriented analysis (Unified Process)(II) Describe Responsibilities Describe Attributes and Associations Qualify Analysis Mechanisms Tutorials	
(6)	- Object-oriented analysis (Unified Process)(III) Describe Responsibilities Describe Attributes and Associations Qualify Analysis Mechanisms Tutorials	
(7)	Introduction to Design Patterns	First examination
(8)	Creational Design Patterns	
(9)	Structural Design Patterns (I)	
(10)	Structural Design Patterns (II)	
(11)	Behavioral Design (I)	
(12)	Behavioral Design (II)	
(13)	Introduction to Component Technologies	Second examination
(14)	Component-Design(I)	
(15)	Component-Design(II)	Due date: First homework

(16) Final Examination	Revision	
-----------------------------------	-----------------	--

age 4 of 4

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

1. Design Patterns, Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides., Addison-Wesley Professional, 1995
2. Software Architecture in Practice, Paul Clements, and Rick Kazman, Addison-Wesley, 2003