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

Course Syllabus

Course Title: Software Modeling	**Course code: 0721222**
Course Level: 2	**Course prerequisite: 0721210**
Lecture Time: 13.10 – 14.00	**Credit hours: 3h +1h (b practice)**

Academic Staff Specifics

Name	Rank	Office Number and Location	Office Hours	E-mail Address
Maouche Mourad	**Associate Professor**	**Room 602 IT building**	STT:10.00-11.00 MW: 11.00-12.00	mmaouch@philadelphia.edu..jo

Course description:

Models are an integral part of every engineering discipline, as they become in software engineering. Modeling can be done in many ways and with different levels of formality. Modeling concerns software products, software processes, enterprise and business processes. Teaching modeling is as important as the modeling itself since it educates the future software engineers in good modeling practices.

Course objectives:

The aim of this course is to:

- Introduce early in the Software Engineering curricula the concept of modeling, its motivations and benefits (lectures).
- Study several modeling frameworks in Software Engineering (lectures).
- Study UML constructs.
- Provide good modeling practices through various real case studies (tutorials, workshops, laboratory).

Course components:

Texbooks

1. Object-Oriented Systems Analysis and Design Using UML, Simon Bennett and Steve McRobb and Roy Farmer, MacGraw-Hill, 2010
2. Object-Oriented Analysis and Design for Information Systems: Modeling with UML, OCL, and IFML,, Raul Sidnei Wazlawick. Elsevier, 2014

Support material (s): slides, books, software modeling tools

Teaching methods: Lectures, Tutorials, Laboratory

Duration: 16 weeks, 64 hours in total. Lectures: 42 hours, Tutorial: 8 hours. Laboratory: 14 hours

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 modeling 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 modeling perspective/views (structure, behavior, interaction) (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)
7. Distinguish General purpose modeling languages from Domain specific modeling language.(A2)

Cognitive skills (thinking and analysis)

8. Build and design software models. (B1)
9. Apply abstraction and separation of concerns principles to build software model. (B5)
10. Analyze, transform, improve, validate models. (B2)
11. Evaluate software models. (B4)

Practical skills

12. Draw software Models using appropriate software modeling tools (C2)
13. Practice specific modeling languages (C3).
14. Prepare and deliver coherent and structured verbal and written technical report.(C7)
15. Use the scientific literature effectively. (C8)

Transferable skills

16. Solve problems (D3)
17. Use creativity (D2)
18. Communicate effectively with non-specialist as well as computer scientist, (D4)

Learning outcomes achievement

- **Development**: A1, A2 are developed through Lectures and Tutorials.

  B1, B2, B4, B5, D2, D3 are developed through Tutorials and Assignments.
  C2, C3, C7, and C8 are developed through practical laboratory sessions.

- **Assessment** : A1, A2, B1, B2, B4, B5 and D3 are assessed through Quizzes, written exams, and Assignments.

  C2, C3, C7, C8, and D3 are assessed through practical assignment examinations

.

Assessment instruments

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

Allocation of Marks	
Assessment Instruments	**Mark**
First examination	**20**
Second examination	**20**
Final examination:	**40**
Classworks/Practice	**20**
Total	**100**

Documentation and academic honesty

- Documentation style (with illustrative examples)
  - Practical works reports must be presented according to the style specified in the homework and practical work guide
- Protection by copyright
- Avoiding plagiarism
  - Any stated plagiarism leads to an academic penalty

Course/module academic calendar

week	Basic and support material to be covered	Homework/reports (due dates)
(1)	**Introduction to software modeling**: **Overview on UML**	
(2)	**Structural Diagram**: class/object, Data Type, Object Diagrams, Class diagrams	
(3)	Tutorial1 **Structural Diagram**: Associations	**Classwork1**
(4)	Tutorial2 **Structural Diagram**: Composition/Aggregation	
(5)	Tutorial3 **Structural Diagram** : Inheritance	

(6) **First Examination**	Tutorial4 **Behavior/Interaction Model:** Use case Modeling	**Classwork2**
(7)	Tutorial5 **Behavior/Interaction Model**: Use Case Refinement	
(8)	Tutorial6 **Behavior/Interaction Modeling**: Use case Relationships	
(9)	Tutorial7 **Behavior/Interaction modeling:** Use cases – Relationships	**Classwork3**
(10)	Tutorial7 **Behavior Diagram**: Activity Diagrams	
(11) **Second Examination**	**Behavior Diagram:** Activity Diagrams Tutorial8	
(12)	**Behavior/Interaction Diagrams:** Sequence Diagram Tutorial9	**Classwork 4**
(13)	**Behavior/Interaction Diagrams**: Sequence diagrams	
(14)	**Behavior/Interaction Diagrams**: Sequence diagrams Tutorial10 **Behavior Diagrams**: State diagram	**Classwork 5**
(15)	**Behavior Diagrams**: State diagram Tutorial11	**Classwork6**
(16) **Final Examination**	**Revision**	

Expected workload

On average students need to spend 2 hours of study and preparation for each 50-minutes 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

1. Object Oriented System Analysis and Design Using UML. J. Mala, S. Geeta, MacGraw-Hill, 2013

2. Object oriented Software Engineering; Using UML, Patterns and Java; B. Bruegge, A. H. Dubois; New Delhi: Dorling Kindersley/Pearson, Second Edition, (Eleventh impression 2012)

3. Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures; Hassan Gomaa, Cambridge University Press, 2011
4. Modeling software behavior; a craftsman's approach, Jorgensen, Paul C, CRC. Press Online, 2009
5. . The Art of Software Modeling, Benjamin A. Lieberman, Auerbach Publications, 2006.

6. Object Oriented Modeling and Design with UML, M. Blaha, J. Rumbaugh, second edition, Pearson, Prentice Hall, 2005