Philadelphia University
Faculty of Administrative and Financial Sciences
Department of Business Networks and System Managements

	Course Syllabus
Course Title: Software Engineering	**Course code: 0371432**
Course Level: 4	**Course prerequisite (s) and/or co-requisite (s): 371211**
Lecture Time:	**Credit hours: 3**

		Academic Staff Specifics		
Name	**Rank**	**Office Number and Location**	**Office Hours**	**E-mail Address**
Ahmad Al Ghoul	M.Sc.	**32404 / Admin. Buld.2**		**Ahmad4_2_69@hotmail.con**

Course module description:

This is a course to introduce theory and practice for software engineering in terms of software development that includes software requirement specifications, design, implementation, and testing. Students will undertake a team-based project. A team-based project working in small groups addresses requirements analysis and specification, software architecture, detailed design, implementation, and testing through a relatively complex software system. The Unified Modeling Language (UML) is used to specify requirements and design the system.

Course module objectives:

This module represents the third semester of an imperative-first introductory track that covers the fundamental of Software Engineering concepts. It builds the fundamental concepts of project management including scheduling, risk analysis, and cost estimation, Aswell as, Unified Modeling Language including the use case model, class model and collaboration model.

Course/ module components

- **Books**

Text book:

Software Engineering *(8th or 9th Edn), Ian Sommerville, Addison Wesley,*

Homework guide.

Homework is an essential part of the educational process.
The homework in this course will reinforce the material covered in the classroom and provide time for practice. Students will earn points for each homework assignment completed. Homework assignments will be graded based on completion.

Teaching methods:

Lectures, discussion groups, tutorials, problem solving, debates, etc.
20% continuous assessments
10% class participation
70% written examination

Learning outcomes:

After completing the course, the students will be able to:

- Understand the basic concepts of project management including scheduling, risk analysis, and cost estimation
- Understand the differences among software development process models
- Use the Unified Modeling Language including the use case model, class model and collaboration model.
- Identify functional and non-functional requirements for software systems
- Specify software requirements using the use case model
- Analyze static and dynamic aspects of software requirements using the class model and collaboration model
- Design software architecture in terms of components and interactions between the components
- Understand the differences among basic software architectural patterns
- Make detailed design including operation interface design and relational database schema derivation
- Make differences between software inspection and testing in Validation and Verification
- Understand how to make test cases for software testing
- Know the assurance approaches for critical systems
- Understand maintenance of software systems through software re-engineering

Communication skills (personal and academic).

Assessment instruments

- Short reports and/ or presentations, and/ or Short research projects
- Quizzes.
- Home works
- Final examination: 50 marks

Communication skills (personal and academic).

-Module language: English

-For every lecture the last five minutes will be open for discussion. For further discussion, the students are welcome at the lecturer s office hour as appeared in first page.

Time Management: Assignments are varied, integrated, and overlapping, and students must focus on multiple issues, projects, and demands. Students must, therefore, take responsibility for planning and pacing their own work as well as developing time management skills.

Project Development: Groups of approximately four to six students develop projects, complete research, schedule meetings, write papers and reports, and deliver a 20-30 minute oral presentation using visual aids.

Group Management: Students work on group projects to practice interpersonal skills by communicating with group members, other groups, and peers outside the group.

Allocation of Marks	
Assessment Instruments	**Mark**
First examination	**20**
Second examination	**20**
Final examination:	**40**
Reports, research projects, Quizzes, Home works, Projects	**20**
Total	**100**

Documentation and academic honesty

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

Definition of Plagiarism

Plagiarism is the unacknowledged borrowing of another writer's words or ideas.

How Can Students Avoid Plagiarism?

To avoid plagiarism, you must give credit whenever you use
another person's idea, opinion, or theory;

- any facts, statistics, graphs, drawings—any pieces of information—that are not common knowledge;
- quotations of another person's actual spoken or written words; or
- Paraphrase of another person's spoken or written words.

If you are in doubt about whether what you are doing is inappropriate, consult your instructor. **A claim that "you** didn't know it was wrong" will not be accepted as an excuse.

Penalty for Plagiarism

The minimum penalty for an act of plagiarism is a 0 on the assignment, homework, and project. Serious cases of plagiarism may result in failure in the course as a whole, or expulsion from the university.

Course/module academic calendar

week	Basic and support material to be covered	Homework/reports and their due dates
(1)	Introduction to Software Engineering - Chapter1	
(2)	Computer-Based System Engineering - Chapter2	
(3) - (4)	Software Processes - Chapter3	Report 1: proposal project
(5)	Project Management - Chapter4	
(6) - First Exam	Software Requirements - Chapter5	
(7)	Requirements Engineering	

	Processes - Chapter6	
(8)	System models - Chapter7	Drawing using Ratioal Rose
(9)	Software Prototyping - Chapter8	Report 2: Prototype project
(10)	Software Design - Chapter10	
(11) - Second Exam	Verification and Validation - Chapter19	
(12)	Testing - Chapter20	
(13) - (14)	Quality Managements - Chapter24	Presentation
(15) - Specimen Exam **(16) - Final Exam**	Case Study	

The course content is tentative and subject to change according to time left in the semester and students' comprehension of course material.

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

- *RS Pressman.* ***Software Engineering: A Practitioner's Approach.*** *McGraw-Hill,2003.*
- *G Booch, J Rumbaugh, and I Jacobson.* ***The Unified Modelling Language User Guide.*** *Addison Wesley, 2005*
- *Stevens P. with Pooley, R.* ***Using UML: Software Engineering with Objects and Components,*** *(Updated edition, 2006) Addison-Wesley,*

Websites

www.softwareEngeneering.com