Future Academy
Higher Future Institute for Specialized Technological Studies

Course Specification

1- Course information:

Course Code:	CSC441
Course Title:	Software Engineering
Year/level	4th
Academic Programs	Computer science (B.Sc.)
Contact hours/ week	(Theoretical=2hrs, Practical/Tutorial=2hrs), Total=4hrs

2- Course aims:

- *Learn the major alternative methodologies used in developing software.*
- *Analyze the business requirements for information and to develop an appropriate strategy to provide the required information service.*
- *Use various information gathering techniques for eliciting user information requirements and system expectations.*
- *Construct and interpret a variety of system analysis and design models including UML diagrams and structured models.*
- *Produce the required systems documentation including project plan at each point in the software development*

3- Intended learning outcomes of the course (ILOs):

a- Knowledge and understanding:

On successful completion of this course, the student should be able to:
a1- Define a comprehensive understanding of core software engineering principles.
a2- Determine the system requirements.
a3- Present principles of robust and maintainable software systems.
a4- List software development tools effectively and the role of emerging technologies in software engineering.
a5- Stat various testing techniques to ensure software quality and the importance of software quality assurance.

b- Intellectual skills:

On completing this course, the student should be able to:

b1- Create detailed and accurate software requirements specifications and functional and non-functional requirements.

b2- Formulate software development methodologies (e.g., Agile, Waterfall), clear and concise software designs, and modeling software systems using appropriate techniques (e.g., UML).

b3- Design testing and acceptance strategies for software engineering systems.

b4- Apply appropriate programming languages and tools for clean, efficient, and maintainable code.

b5- Review software components and systems to ensure quality.

c- Professional and practical skills:

At the end of this course, the student will be able to:

c1- Show software development methodologies (e.g., Agile, Waterfall) to manage projects effectively.

c2- Analyze graphical techniques to create conceptual system models of various types.

c3- Label aims and objectives for a project and develop metrics to assess them.

c4- Apply ethical principles in software development, such as data privacy and responsible software usage.

c5- Recommend involvement with developers, designers, and stakeholders to achieve project goals.

d- General and transferable skills:

On successful completion of this course, the student should be able to:

d1-Management complex problems and break them down into smaller, manageable subproblems.

d2- Compute potential challenges and risks associated with a project to mitigate them.

d3- Communicate technical concepts clearly and concisely to both technical and non-technical audiences.

d4- Working in groups with team members to achieve common goals.

d5- Think creatively for innovative solutions to problems and commitment to lifelong learning and professional development.

4- Course contents

Week No.	Topics/units	Number of hours		ILO's
		Lecture hours	Practical/ Tutorial hours	
1	Introduction to software engineering, Professional software development	2	2	a1, c1
2	what is meant by software engineering and Software engineering ethics.	2	2	a1, c4
3	Software process models, Process activities and Process improvement.	2	2	a3, c1
4	Coping with change + Quiz 1	2	2	d2
5	Agile methods, Agile development techniques, Agile project management, Functional and non-functional requirements and Requirements.	2	2	b1, c1
6	Engineering processes, Requirements elicitation, Requirements specification,	2	2	a2, b1

	Requirements validation and Requirements change			
7	Midterm Exam	-	-	-
8	System modeling: Context models, Interaction models, Structural models, Behavioral models and Model-driven engineering	2	2	b2, c2
9	Architectural design decisions, Architectural views, Architectural patterns and Application architectures	2	2	a3, b2
10	Design and Implementation: Object-oriented design using the UML, Design patterns, Implementation issues and Open-source development	2	2	b2, b4
11	Software development testing, Test-driven development + Quiz 2	2	2	a5, b5
12	Release testing And User testing	2	2	a5, b5
13	Software Evolution processes	2	2	c3, a3
14	Legacy systems and Software maintenance	2	2	c3, a3

5- Teaching and learning methods

Methods	ILO's																			
	a1	a2	a3	a4	a5	b1	b2	b3	b4	b5	c1	c2	c3	c4	c5	d1	d2	d3	d4	d5
Lectures	√				√															
Tutorial / Practical sections					√					√	√				√					
Self-learning																				√
Assays and reviews						√				√										
Discussion groups																		√	√	
Brainstorming										√							√			√
Blended-learning																				
E-learning					√													√	√	√

6- Teaching and learning methods for Low-achieving students

- Extra teaching hours for those who need help
- More quizzes to assess their ability for understanding the course
- Encourage the team work for those students with other advanced ones to increase their participation and understanding

7- Student assessment

Assessment method	Time	Grade weight (%)	Week	ILOs
Course Work (Tutorial Exercise and Assignments)	Through the semester	20	Every week	a3, a4, a5, b2, b5, c2, c3, d1, d2, d4
Quiz 1	30 mins	5	Week 4	a1, a2, b1, d3
Mid-term exam	60 mins	10	Week 7	a1, a3, a5, b1, b3, c3, d2, d4
Quiz 2	30 mins	5	Week 11	
Final Written exam	120 mins	60	End of the semester	

8-List of references

8.1. Student notebooks:

- Comprehensive instructor notes ("PowerPoint slides") are available on the course web page ("Google Classroom")

8.2. Essential textbooks:

- **SOFTWARE ENGINEERING - Ten Edition - Ian Sommerville.**

8.3. Recommended textbooks:

- **Roger S Pressman - Software Engineering-A Practitioner's Approach - 8th Edition.**

8.4. Journals, Periodical and Reports ……..etc.

- **Martin, R. C. (2009).** *Clean code: a handbook of agile software craftsmanship*. **Pearson Education.**

- **Gamma, E. (1995). Design patterns: elements of reusable object-oriented software.**

- **Jalote, P. (2008).** *A concise introduction to software engineering*. **Springer Science & Business Media.**

- **Dooley, J. F., & Kazakova, V. A. (2024). Introduction to Software Development. In** *Software Development, Design, and Coding: With Patterns, Debugging, Unit Testing, and Refactoring* **(pp. 1-7). Berkeley, CA: Apress.**

8.5. Websites

- **https://web.gisma.com/beng-software-engineering?utm_source=google&medium=cpc&utm_campaign=gisma_hochschule_be_software_engg_tier2_search&utm_term=be%20software%20engineering&creative=646519426972&network=g&device=c&placement=&adposition=&gad_source=1&gclid=CjwKC**

- **https://www.edx.org/learn/software-engineering**
- **https://www.coursera.org/courses?query=software%20engineering**
- **https://www.codecademy.com/resources/docs/general/software-engineering**

Course Coordinator: *Dr. Ahmad Nagm*
Head of department: *Prof. Dr. Yasser F. Ramadan*
Date of Approval: 24/7/2024