GAU, Faculty of Engineering

Course Unit Title	Fundamentals of Software Engineering
Course Unit Code	SEN201
Type of Course Unit	Compulsory for SE
Level of Course Unit	2nd year BSc
National Credits	3
Number of ECTS Credits Allocated	5
Theoretical (hour/week)	3
Practice (hour/week)	0
Laboratory (hour/week)	0
Year of Study	2
Semester when the course unit is delivered	3
Mode of Delivery	Face to face, E-learning
Language of Instruction	English
Prerequisites	SEN101
Corequisites	
Recommended Optional Programme Components	

Objectives of the Course:

- Introduce the fundamental principles, processes, and practices of software engineering
- Present software development life cycle models, agile methods, and teamwork practices
- Develop student ability to produce basic software engineering artifacts and documentation
- Encourage students to relate engineering discipline to real-world software development

Learning Outcomes

When this course has been completed the student should be able to		Assess.
1	Learn how to explain the role of software engineering in developing reliable and maintainable systems	1,3
2	Learn to compare software process models and agile development approaches	1,3
3	Learn how to prepare basic software engineering documents including requirements and design artifacts	1,3
4	Learn how to collaborate in teams using version control and structured development practices	1,3
5	Understand the fundamentals of software quality, maintenance, and professional engineering practice	1,3
6	Learn how to develop a small software project by applying software engineering principles	1,3

Assessment Methods: 1. Written Exam, 2. Assignment, 3. Project/Report, 4. Presentation, 5. Lab Work

Course's Contribution to Program

		CL
1	Ability to understand and apply knowledge of mathematics, science, and engineering	3
2	Ability to design and conduct experiments as well as to analyze and interpret	2
3	Ability to work in multidisciplinary teams while exhibiting professional responsibility and ethical conduct	3
4	Ability to apply systems thinking in problem solving	3
5	Knowledge of contemporary issues while continuing to engage in lifelong learning	2
6	Ability to use the techniques, skills and modern engineering tools necessary for engineering practice	3
7	Ability to express their ideas and findings, in written and oral form	3

8	Ability to design and integrate systems, components or processes to meet desired needs within realistic constraints	2
9	Ability to approach engineering problems and effects of their possible solutions within a well structured, ethically responsible and professional manner	2

CL: Contribution Level (1: Very Low, 2: Low, 3: Moderate, 4: High, 5: Very High)

Course Contents

Week	Chapter	Subject	Exams
1	1	Introduction to Software Engineering	
2	2	Software Process Models	
3	3	Agile Development and Scrum	
4	4	Requirements Engineering Overview	
5	5	Modeling and UML Basics	
6	6	Software Design Fundamentals	
7	7	Implementation Practices and Version Control	
8			Midterm
9	8	Testing and Quality Basics	
10	9	Software Maintenance and Evolution	
11	10	Project Management Fundamentals	
12	11	Teamwork, Ethics, and Professional Practice	
13	12	Mini Project Development and Presentation	
14	13	Course Review	
15			Final

Recommended Sources

- Textbook: Software Engineering, 10th Edition, Ian Sommerville Supplementary: Software Engineering: A Practitioner's Approach, 9th Edition, Roger S. Pressman and Bruce R. Maxim Supplementary: Essentials of Software Engineering, 4th Edition, Frank Tsui, Orlando Karam, and Barbara Bernal

Assessment

Midterm	30 %
Final exam	40 %
Project	20 %
Quizzes	10 %

ECTS Allocated Based on the Student Workload

Activities	Number	Duration (hour)	Total Workload (hour)
In-class lecture (including exam weeks)	15	3	45
Midterm exam	1	1.5	1.5
Midterm exam preparation	1	15	15
Final exam	1	1.5	1.5
Final exam preparation	1	20	20
Quiz	3	2	6
Assignment	4	4	16
Project/presentation/report writing	1	12	12
Lab and tutorial	0	0	0
Self-study	15	2	30
Total Workload			147.00

Total Workload / 30 (h)	4.90
ECTS Credit of the Course	5