The University of Jordan
School of Engineering
Department of Computer Engineering
Fall Term 2020/2021

Course Title

Software Engineering and ethics 0907742 (3 Cr. – Core Course)

Catalog Description

Fundamental areas of software engineering including life-cycle-paradigms, metrics and tools. Information hiding architecture, modular languages, incremental programming, and very high level languages. Pragmatic aspects of the production of software systems, dealing with structuring principles, design methodologies and informal analysis. Emphasis is given to development of large, complex software systems. Embracing the DevOps Culture: DevOps is a culture of transforming enterprises towards continuous agile delivery in a faster and smarter way, and to make solutions and processes automated. Term project.

Prerequisites by Course

None

Prerequisites by Topic

Students are assumed to have a background in the following topics:

- Object-oriented programing
- Data structures

Textbook

Sommerville, Ian. Software Engineering, Tenth Edition.

References

- Meyers, Scott. Effective C++, Third Edition.
- McConnell, Steve. Code Complete, Second Edition.
- UML and the Unified Process - Jim Arlow, Ila Neustadt. Addison Wesley. 2nd Edition, 2005
- Current literature found in trade journals, professional-society publications, manufacturer's publications, etc., related to the course learning objectives.

Schedule & Duration

15 Weeks, 31 lectures, 75 minutes each

.

Minimum Student Material

Textbook, class handouts, some instructor keynotes, and any additional reading assigned by the instructor.

.

Minimum College Facilities

Classroom with whiteboard and projection display facilities, library, and computer laboratory.

Course Objectives

The content starts with development lifecycle models, such as agile development, and then continues to cover requirements specification, the Unified Modelling Language (UML), software architecture, object-oriented analysis and design, design patterns and testing.

Software engineering is an inherently practical subject and applying the concepts being taught is a vital component of developing expertise in this area. Consequently, students undertake a substantial group project, working through a number of stages of the development of a larger software application. Students will be supervised, but will be expected to largely organize themselves and their work, learning key transferrable skills in management and organization.

Course Learning Outcomes

and Relation to ABET

Learning Outcomes

Upon successful completion of this course, a student should:

1. Appreciate the wider engineering issues that form the background to developing complex and evolving software-intensive systems. [1,2]
2. Plan and deliver an effective software engineering process, based on knowledge of widely used development lifecycle models. [2]
3. Capture, document and analyze requirements. [2,4]
4. Employ group working skills including general organization, planning and time management and inter-group negotiation. [2,4]
5. Translate a requirements specification into an implementable design, following a structured and organized process. [2]
6. Make effective use of UML, along with design strategies such as defining a software architecture, separation of concerns and design patterns. [2]
7. Formulate a testing strategy for a software system, employing techniques such as unit testing, test driven development and functional testing. [2]
8. Evaluate the quality of the requirements, analysis and design work done during the module. [2]

Course Topics

		Hrs
1.	Software Engineering Principles	6
2.	Requirements Engineering	9
3.	Introduction to UML	9
4.	The Analysis and Design Process	9
5.	Design Principles	6
6.	Testing	6

Policies

- Attendance is required. Class attendance will be taken every class and the university polices will be enforced in this regard.
- All submitted work must be yours
- Cheating will not be tolerated
- Open-book exams
- Join the Facebook group of this course
- Check department announcements at: http://www.facebook.com/pages/Computer-Engineering-Department/369639656466107 for general department announcements.

Assessments

Exams, Quizzes, Reports, and Presentations

Grading Policy

Quizzes	20%
Midterm Exam	30%
Final Exam	50%
Total	**100%**

Program Learning Outcomes

i Demonstrate a sound, in-depth and up-to-date technical knowledge in the field of specialization.

ii Ability to identify and solve engineering problems in their chosen field of study.

iii Acquire the skills for continued professional development and independent self-study.

iv Demonstrate the ability to communicate technical information effectively and professionally both orally and writing.

Last Updated

February 11 2021