Zarqa University
Faculty of Information technology
Department: Software engineering
Course title: Large-scale Software Development

Instructor:
Lecture's time:
Semester:
Office Hours:

Course description:

Large-scale software implementations of the hierarchy of engineering analysis, design, and decision evaluation. Computer-aided engineering programs with state-of-the-art computer tools and methods. Operator overloading, dynamic polymorphism, graphical user interfaces, generic programming, dynamic link libraries, and multiple threads.

Aims of the course:

1. learn about large-scale software implementation
2. Gain first-hand experience of large-scale software analysis, design and decision evaluation
3. Practice technical discussions with your classmates, and practice finding relevant resources on the Internet
4. Understand the scope of a typical software product and the people and processes that are needed to produce it. Start thinking about where you fit into the software engineering profession.

Intended Learning Outcomes: (ILOs)

A. KnowledgeandUnderstanding

A1.ConceptsandTheories:

Provide the students with the basic and advanced uses of computer-aided engineering programs with state-of-the-art computer tools and methods

Provide the students with required skills of understanding and knowledge of how to implement large-scale software

A2.Contemporary Trends, Problems and Research:

A3.Professional Responsibility:

Teaching the students the problems they will face and how to skip these problems to prepare them to be responsible for developing large-scale software

B. Subject-specific skills

B1. Problem solving skills:

Learn how to use different tools and methods of software development to be prepared for any problems they might face during software development stages

B2.ModelingandDesign:

Lear how to build up and design software using the appropriate tools

B3.ApplicationofMethodsandTools:

Learn how, when and where to use the appropriate tools, methods and techniques

C. Critical-Thinking Skills

C1.Analytic skills: Assess

Distinguish between different toolsand methods uses in a specific software

C2.Strategic Thinking:

Structure, using functional programming, large programs so that they become easier to understand and manage, identify the need of design patterns in development of new or in administration of existing code, as well as implement these where appropriate,

C3.Creative thinking and innovation:

Design and document public APIs with a clear responsibility,
Design entities so that they become testable and write tests for them,
Protect internal design of a program from the public APIs,
Develop a programme in collaboration with others,
Review and reflect on source code,

Develop given source code

D. General and Transferable Skills (other skills relevant to employability and personaldevelopment)

D1. Communication:

Communicate with the students in the proper way to deliver the required skills and provide them with necessaryknowledge about software development tools and methods.

Understand and master the parameters in software development that make source code sustainable, re-usable

and flexible during changing requirements.

D2.Teamwork and Leadership:

Divide class students into a number of groups to teach them how to work in a teamwork and providing them with assignments and home works to discuss the uses of different tools and methods of software development

Course structures:

Week	Credit Hours	ILOs	Topics	Teaching Procedure	Assessment methods
1-4	3	A1, A3, B1, B2, B3, C1, C2 and C3	Introduction and Programmes as compositions of functions	Theoretical and practical methods	Testing students level of understanding and areas of weaknesses by different methods such as class participation, quizzes, class homework, exams and activity file

5	1	D1 and D2	First exam and Assignment	Practical	
5-9	3	A1, A3, B1, B2, B3, C1, C2 and C3	Comparison between functional and object-oriented programming	Theoretical and practical methods	Testing students level of understanding and areas of weaknesses by different methods such as class participation, quizzes, class homework, exams and activity file
10-11	3	A1, A3, B1, B2, B3, C1, C2 and C3	Handling and identification of code complexity.	Theoretical and practical methods	Testing students level of understanding and areas of weaknesses by different methods such as class participation, quizzes, class homework, exams and activity file
11	1	D1 and D3	Second exam and assignment	Practical	
11-12	3	A1, A3, B1, B2, B3, C1, C2 and C3	The open-closed principle, and Modelling of information.	Theoretical and practical methods	Testing students level of understanding and areas of weaknesses by different methods such as class participation, quizzes, class homework, exams and activity file

12-14	3	A1, A3, B1, B2, B3, C1, C2 and C3	Business logic and rules compared with handling of state, and APIs. Documentation of entities.	Theoretical and practical methods	Testing students level of understanding and areas of weaknesses by different methods such as class participation, quizzes, class homework, exams and activity file
15	3	A1, A3, B1, B2, B3, C1, C2 and C3	Testability and the importance of tests, Tests as documentation. Mutability and persistent data structures	Theoretical and practical methods	Testing students level of understanding and areas of weaknesses by different methods such as class participation, quizzes, class homework, exams and activity file
15	2	D1 and D3	Final Exam	Theoretical and practical methods	

References:

A. Main Textbook:

- Required: Sommerville, Ian. Software Engineering. 6th edition.
- Required: Brooks, Frederick P. The Mythical Man Month. Any edition.
- Required: Fowler, Martin. UML Distilled. 2nd edition.

Assessment Methods:

Methods	Grade	Date
Activity file including assignments, quizzes, reports, etc.	10%	
1st exam	20%	
2nd exam	20%	
Final exam	50%	