Arab Academy for Science and Technology and Maritime Transport
Computer Science Curriculum
Course Syllabus

Course Code: SE491	**Course Title:** Software Component Design	**Classification: E**	**Coordinator's Name:** Dr. Abeer Bader	**Credit Hours:** 3
Pre-requisites: SE291 (Introduction to Software Engineering)	**Co-requisites:** None	**Schedule:** Lecture: 2 hours Tutorial-Lab: 2 hours		

Course Description:

This course begins with design fundamentals, including concepts, context and processes. It then progresses through key design issues; structure and architecture; user interface design; design quality analysis and evaluation; notations; strategies and design tools, all within the context of real-world challenges.

Software design is both the process of defining the architecture, components, interfaces and other characteristics of a system, and the result of that process. Essentially, software design is the software life cycle activity in which parameters outlined and defined in the Requirements process are translated into a description of a software system's internal structure that can be used as a basis for construction.

Textbook:
J. M. Smith, *Elemental Design Patterns*, Addison-Wesley Professional; 1st edition (April 7, 2018).

References:

- Michael R. Blaha, and James R. Rumbaugh, Object-Oriented Modeling and Design with UML, Pearson; 2nd edition (December 9, 2014)
- Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software Engineering, Addison-Wesley.

Course Objective/Course Learning Outcome:	**Contribution to Program Student Outcomes:**
1. Understand and apply object-oriented design patterns. 2. Understand the use of UML in the design process.	(SO1) Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions. (SO2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
1. Design and implement small software components and systems. 2. Use computer-aided software engineering (CASE) tools in the design process.	(SO2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline. (SO6) Apply computer science theory and software development fundamentals to produce computing-based solutions.
3. Perform independent research on software design. 4. Document software design concepts in a written report. 5. Communicate software design concepts in a brief oral presentation.	(SO3) Communicate effectively in a variety of professional contexts. (SO4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.
Course Outline: 1. Revision of Object Oriented principles 2. Introduction to pattern design 3. Singleton, Factory, and Abstract 4. Factory Design Pattern 5. Builder, Prototype, Adapter and 6. Bridge Design patterns 7. Filter, Composite, Decorator and 8. Façade Design Patterns 9. Proxy, Chain of responsibility, 10. Command and Interpreter Design Patterns	11. Observer, Strategy, Template Visitor and Data Access Object Design pattern. 12. Component-based software architecture overview and Principles of Component-Based Design 13. JavaBean Component 14. Enterprise JavaBean EJB 15. Web Services Components WSC 16. Project Demos
