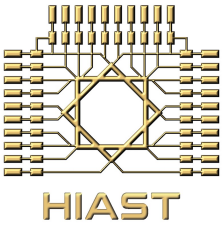
Syrian Arab Republic

Higher Institute for Applied Sciences and Technology



Course Specification Document

Title	Software Design

Credits	5 ECTS

Aims	This course aims to enable the student to understand the concepts related to software design,the engineering principles on which it is based and its different levels (architecture design, software design, ...) by adopting an object oriented approach and using UML and design patterns. This enhances abstraction and engineering design mindset in the student and focuses on his role as a solution designer rather than just a developer.

Intended learning outcomes

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

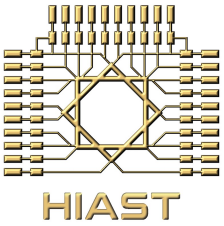
- Recognize the concept of architecture and its most common types.
- Understand the fundamental concepts related to object-oriented analysis and design.
- Identify general design patterns for assigning responsibilities to classes (GRASP) and design methods based on them to achieve high-quality designs.
- Recognize the factors influencing software design.
- Comprehend the design background of software solutions and available alternatives.
- Identify code design patterns based on the fundamental principles that contribute to achieving the highest standards in software quality.
- Select the best design alternatives that meet the highest standards of software quality after analyzing functional and non-functional project requirements.
- Conduct objective comparisons between the features offered by programming languages and frameworks.
- Develop agile applications through the optimal utilization of data structures, programming languages, and frameworks within architectural structures.

Syllabus

- **Design techniques:** Contracts, interaction diagrams in UML (collaboration diagrams and sequence diagrams).
- **Software design principles:** GRASP design patterns, realizing use cases and creating collaboration/sequence diagrams from contracts.
- **From design to implementation:** Generating codes from class and sequence diagrams.
- **Programming principles:** DRY, KISS, SOLID.

Syrian Arab Republic

Higher Institute for Applied Sciences and Technology



- **Concept of design patterns:** Definition of design patterns, functions of design patterns, components of design patterns, types of design patterns.
- **Creational design patterns:** Abstract Factory, Builder, Factory Method, Prototype, Singleton.
- **Structural design patterns:** Adapter, Bridge, Composite, Decorator, Façade, Flyweight, Proxy.
- **Behavioral design patterns:** Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Momento, Observer, State, Strategy, Template Method, Visitor.
- **Architectural Patterns - Introduction:** Concept of software system architecture and its relationship with design principles and patterns, Monolith Applications, a reminder of tier patterns, MVC.
- **Architectural Patterns:** DDD (Domain Driven Development), Clean Architecture.
- **Architectural Patterns – Distributed Systems:** Concept of distributed systems, communication between systems (Sync/Async), web services, message brokers.