Architecting Clean Code From SOLID to Design Patterns

Workshop Overview

Item	Details
Course Title	*Architecting Clean Code: From SOLID to Design Patterns*
Duration	8 Sessions (Total: 24 hours)
Format	Practical, project-based training program
Target Audience	FCAI Students — Intermediate to advanced developers
Organized By	DSC Cairo University Chapter
Instructor(s)	Omar Betawy, Bassel Ahmed
Venue	Creativa
Tools Used	VS Code / IntelliJ / PyCharm, GitHub, UML Diagrams, ChatGPT (for refactoring assistance)
Goal	Equip students with strong architectural thinking, applying SOLID principles and design patterns to write scalable, maintainable, and professional-grade software.

Timeline:

Session	Theme & Topics	Date	Time
Session 1 – "Writing for the Future" (Foundations of Clean Code)	• What is Clean Code & why it matters • Code smells & refactoring basics • Cohesion, coupling, and maintainability • Introduction to SOLID & design principles	**4 Nov**	11:00–03:00 PM
Session 2 – "Breaking the Monolith" (SRP & OCP)	• Single Responsibility Principle (SRP) • Open/Closed Principle (OCP) • Refactoring large classes • Extending behavior without modifying existing code	**11 Nov**	11:00–03:00 PM
Session 3 – "Decoupled Thinking" (LSP, ISP, DIP)	• Liskov Substitution Principle (LSP) • Interface Segregation (ISP) • Dependency Inversion (DIP) & Dependency Injection • Refactoring example: Payment or Notification system	**18 Nov**	11:00–03:00 PM
—	**Midterm Exams Break**	**22 – 27 Nov**	—
Session 4 – "From Chaos to Clarity" (Applying SOLID in Real Projects)	• Combining all SOLID principles • Identifying tradeoffs and anti-patterns • Case study: E-Commerce / Library System refactor	**2 Dec**	11:00–03:00 PM
Session 5 – "The Architecture Mindset" (Intro to Design Patterns)	• What are Design Patterns? • Pattern taxonomy (Creational, Structural, Behavioral) • Factory, Singleton, Abstract Factory basics	**9 Dec**	11:00–03:00 PM
Session 6 – "Designing Creatively" (Creational Patterns Deep Dive)	• Builder, Prototype, Abstract Factory in depth • Scenario: UI Theming or Document Builder • Combining multiple creational patterns	**13 Dec**	11:00–03:00 PM

Session 7 – "Building the Skeleton" (Structural Patterns)	• Adapter, Facade, Decorator, Composite, Proxy• Composition vs inheritance in architecture• Mini-project: Implement Decorator or Facade	**16 Dec**	11:00–03:00 PM
Session 8 – "Patterns in Motion" (Behavioral Patterns + Final Project)	• Strategy, Observer, Command, Template Method, Chain of Responsibility• Combining patterns + SOLID• Final project presentations & feedback	**20 Dec**	11:00–03:00 PM

Overall Course Objectives

- Build a strong understanding of software design principles and architecture.

- Enable students to write flexible, maintainable, and scalable code.

- Teach practical application of all SOLID principles.

- Explore 23+ classic design patterns with real-world examples.

- Strengthen teamwork, design communication, and UML diagramming skills.

- Prepare students for professional software engineering, system design interviews, and open-source contributions.

Learning Outcomes (After Completing All Sessions)

By the end of the course, participants will be able to:

- Identify and eliminate code smells through refactoring.
- Apply all five SOLID principles effectively in real projects.
- Recognize, implement, and combine design patterns appropriately.
- Create software architectures that are modular, extensible, and testable.
- Use UML to document and communicate software designs clearly.

- Collaborate on scalable projects following best software engineering practices.

Deliverables & Evaluation

Component	Description	Weight
Weekly Exercises	Refactoring & code design challenges	30%
Mid-course Project	Applying SOLID to a small project	25%
Final Project	Integrating SOLID + Design Patterns into a real-world system	35%
Participation	Code reviews, discussions, teamwork	10%

Certificates

Certificates of Completion will be awarded by **DSC Cairo University Chapter**, **MILSA Program**, and **Faculty of Computing & Artificial Intelligence (FCAI)** — with official approval from the Dean of FCAI.