Software Engineering

Module Description

Module code	CS303-F28	**Credits**	8 ECTS
Programme	Foundation programme	**Level**	Semester 5
Semester	Fall 2028	**Prerequisites**	CS101-F26 Introduction to Informatics; CS103-S27 Data Structures and Algorithms I
Module leader	Dr. Ali Khudiyev	**Email**	ali.khudiyev@wosm.academy
Office hours	Monday–Friday, 10:00–17:00 (Baku time, UTC+4)		

1. Module Overview

Writing a program that works is different from building software that a team can develop, test, ship and maintain for years. Software Engineering is about the second problem. It covers how software projects are organised, how requirements are captured, how systems are designed and kept maintainable, and how quality is assured through testing and review.

The module is practical. Alongside the lectures you will work in a small team on a software project in Java, using the tools and practices of professional development: version control, automated tests, continuous integration and code review. The module closes the core of the Foundation programme and prepares you for the optional research or project work that follows Semester 5.

2. Aims

- Introduce the processes, methods and tools used to develop software systematically.
- Develop your ability to design software that is easy to understand, test and change.
- Teach systematic testing and other quality-assurance practices.
- Give experience of working in a team on a substantial software project.

3. Learning Outcomes

After completing this course, you will be able to:

LO1. Compare software process models, including plan-driven and agile approaches such as Scrum, and choose one suited to a project.

LO2. Elicit and specify requirements as user stories, use cases and acceptance criteria.

LO3. Design software with UML models, design principles such as low coupling, high cohesion and SOLID, and design patterns, and improve existing code by refactoring.

LO4. Test software systematically with unit, integration and system tests, use test-driven development, and measure test coverage.

LO5. Use version control, build automation, continuous integration and code review in a team workflow.

LO6. Plan and deliver a software project in a team, and discuss the professional, ethical and legal responsibilities of software engineers, including open-source licensing.

4. Syllabus

Unit	Topic
1	What is software engineering? Software processes and life cycles
2	Agile methods: Scrum and Kanban
3	Requirements engineering: elicitation, user stories and use cases
4	Version control with Git and collaborative workflows
5	Software modelling with UML
6	Design principles: abstraction, coupling and cohesion, SOLID
7	Design patterns I: creational and structural patterns
8	Design patterns II: behavioural patterns; software architecture styles
9	Clean code and refactoring
10	Software testing I: unit testing and test-driven development
11	Software testing II: integration, system and acceptance testing; coverage
12	Build automation, continuous integration and continuous delivery
13	Code review, static analysis and software quality
14	Software maintenance and evolution; technical debt
15	Project management: estimation, planning and risk
16	Professional practice, ethics and open-source licensing
17	Project presentations & Course Review

5. Teaching and Learning Methods

The module is self-paced. All materials are provided on Moodle. Each unit has a lecture and exercises. Throughout the module you work in a small team on a software project, applying the practices of each unit as you go.

Activity	Hours
Lectures	30
Labs / Tutorials	50
Independent study	150
Total	**230**

6. Assessment

Your knowledge will be assessed with quizzes, the team project and the final exam. The project is assessed on the delivered software and on its requirements, design, tests and development history, together with a short individual reflection. The exam checks whether you can apply design principles, patterns and testing techniques to given problems.

Component	Weight	Due	LOs
Quizzes	15%	Throughout	LO1–LO4
Team project and individual reflection	50%	Throughout	LO2–LO6
Final exam	35%	End of module	LO1–LO4, LO6
Total	**100%**		

7. Reading List

Essential

- I. Sommerville: *Software Engineering*, 10th ed., Pearson, 2016.

Recommended

- E. Gamma, R. Helm, R. Johnson, J. Vlissides: *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley, 1994.
- M. Fowler: *Refactoring: Improving the Design of Existing Code*, 2nd ed., Addison-Wesley, 2018.
- J. Bloch: *Effective Java*, 3rd ed., Addison-Wesley, 2018.
- D. Thomas, A. Hunt: *The Pragmatic Programmer*, 20th anniversary ed., Addison-Wesley, 2019.
- T. Winters, T. Manshreck, H. Wright: *Software Engineering at Google*, O'Reilly, 2020.

8. Materials and Licence

Course materials are published under the Creative Commons Attribution-ShareAlike 4.0 licence unless stated otherwise. See wosm.academy/licence for details.

9. Policies

- **Attendance:** Attendance is not mandatory. The module is self-paced, so you can study the materials when it suits you.
- **Late submissions:** Work submitted after the deadline is penalised.
- **Academic integrity:** All submitted work must be your own. See the Code of Conduct.