



Department of Computer Science

CS3004 – Software Design & Analysis

Fall 2025

Instructor Name: Aamir Raheem

Email address: aamir.raheem@nu.edu.pk

Phone: 111-128-128 X 616

Office Location/Number: New Building

Office Hours: (after your class)

Course Information

Program: BS

Credit Hours: 3

Type: Core

Pre-requisites (if any): Data Structures (course and lab)

Course Description/Objectives/Goals:

Purpose of this course is to enable students to analyze, design and implement software systems and applications. The students learn these skills using the modern object-oriented paradigm, which promises reusable, readable and maintainable software components and applications.

Course Learning Outcomes (CLOs):

At the end of the course students will be able to:

- 1) Explain software development life cycle
- 2) Implement object-oriented principles for software analysis and design
- 3) Use different UML notations for software design
- 4) Develop software design artifacts based on requirements specifications
- 5) Describe software design guidelines and principles

Course Textbook

Timothy C. Lethbridge, Robert Laganaire , Object-Oriented Software Engineering (2nd Edition), McGraw-Hill, 2005

Additional references and books related to the course:

- Blaha and Rumbaugh, Object-oriented Modeling and Design with UML (Second Edition), Pearson Prentice Hall 2005
- Erich Gamma, et al., Design Patterns: Elements of Re-usable Object-oriented Design, Addison-Wesley, 1999
- Martin Fowler, Refactoring: Improving the Design of Existing Code (First Edition), Addison-Wesley
- Robert C. Martin, Clean Architecture: A Craftsman's Guide to Software Structure and Design, Pearson 2017
- Grady Booch et al., Object-Oriented Analysis and Design with Applications (3rd Edition), Pearson 2007.
- Larman, Craig. Applying UML and patterns: an introduction to object oriented analysis and design and iterative development. Pearson Education India, 2012.

Tentative Weekly Schedule

Week	Topics to be covered
1	Intro + SDLC; Characteristics of good programs
2	OOP review, classes and objects; Inheritance, polymorphism, UML, is-a rule
3	Object associations (simple association, aggregation, composition; implementation; one to one, many to one, many to many; reflexive)
4	Multiple inheritance, diamond problem, Java interfaces, C++ virtual inheritance
5	Class diagrams
6	Use case diagrams, use case descriptions
7	Sequence diagrams
8	State diagrams, Activity diagrams
9	Design principles
10	Object-Oriented Metrics
11	Design patterns 1, 2
12	Design patterns 3, 4
13	Design patterns 5, 6
14	Design patterns 7, 8
15	(Miscellaneous topics)

(Tentative) Marks Distribution

1. Class participation (2%)
2. Quizzes (8%)
3. Assignments (10%)
4. Project (10%)
5. Midterm Exams (30%)
6. Final Exam (40%)

Grading Scheme

Absolute grading will be used

Course Policies

1. 80% attendance is required to appear in the exams
2. Plagiarism is not tolerable in any of its form. The penalty may be an 'F' grade in the course. Students bear all the responsibility for protecting their assignments. In case of cheating, both parties will be considered equally responsible!
3. Assignments must be submitted in time. Late submissions (maximum one week) would result in deduction in marks. Only the submitted articles will be marked.