

CS3342: SOFTWARE DESIGN

Effective Term

Semester A 2025/26

Part I Course Overview

Course Title

Software Design

Subject Code

CS - Computer Science

Course Number

3342

Academic Unit

Computer Science (CS)

College/School

College of Computing (CC)

Course Duration

One Semester

Credit Units

3

Level

B1, B2, B3, B4 - Bachelor's Degree

Medium of Instruction

English

Medium of Assessment

English

Prerequisites

CS2310 Computer Programming or CS2311 Computer Programming or CS2334 Data Structures for Data Science or CS3391 Advanced Programming or equivalent

Precursors

Nil

Equivalent Courses

Nil

Exclusive Courses

Nil

Part II Course Details

Abstract

CS3342 Software Design aims to introduce the fundamental Software Engineering design principles and practices of software process and software development methodology. Students will explore techniques to elicit requirements, analyze them, and apply software engineering principles to design their solutions. Additionally, the course also discusses the importance of professional ethics in engineering, equipping students with a strong ethical foundation to navigate the responsibilities of software development. Through a balance of theory and hands-on exercises, students will gain the knowledge and skills necessary to thrive in today's dynamic software engineering landscape.

Course Intended Learning Outcomes (CILOs)

CILOs		Weighting (if app.)	DEC-A1	DEC-A2	DEC-A3
1	Explore the software development lifecycle and explain the structured and object-oriented software development methodologies.		x	x	
2	Identify, analyze and document user requirements for software development.			x	
3	Apply object-oriented analysis using contemporary software design and modeling approaches			x	
4	Design software systems using object-oriented design principles.			x	
5	Discuss software engineering professional ethics.		x	x	

A1: Attitude

Develop an attitude of discovery/innovation/creativity, as demonstrated by students possessing a strong sense of curiosity, asking questions actively, challenging assumptions or engaging in inquiry together with teachers.

A2: Ability

Develop the ability/skill needed to discover/innovate/create, as demonstrated by students possessing critical thinking skills to assess ideas, acquiring research skills, synthesizing knowledge across disciplines or applying academic knowledge to real-life problems.

A3: Accomplishments

Demonstrate accomplishment of discovery/innovation/creativity through producing /constructing creative works/new artefacts, effective solutions to real-life problems or new processes.

Learning and Teaching Activities (LTAs)

LTAs	Brief Description	CILO No.	Hours/week (if applicable)	
1	Lecture	Students will engage in lectures to gain knowledge about software engineering design.	1, 2, 3, 4, 5	3 hours per week
2	Tutorial	Students will engage in tutorial activities to extend their use of computer software tools for software design and modeling.	1, 2, 3, 4, 5	1 hour per week

3	Group projects / Assignment	Students will participate in groups to consolidate their learning as they produce the final group report. Students will apply concepts and skills learned, and discuss their solutions with their group members.	1, 2, 3, 4, 5	After class
---	-----------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------	-------------

Assessment Tasks / Activities (ATs)

	ATs	CILO No.	Weighting (%)	Remarks ("- " for nil entry)	Allow Use of GenAI?
1	Assignment	1, 2, 3, 4	15	Need to acknowledge	Yes
2	Mid-term	1, 2, 3, 4	15	-	No
3	Group Project	1, 2, 3, 4, 5	20	Need to acknowledge	Yes

Continuous Assessment (%)

50

Examination (%)

50

Examination Duration (Hours)

2

Minimum Continuous Assessment Passing Requirement (%)

40

Minimum Examination Passing Requirement (%)

30

Additional Information for ATs

For a student to pass the course, at least 40% of the maximum mark for the continuous assessment and 30% of the maximum mark for the examination must be obtained.

Assessment Rubrics (AR)**Assessment Task**

Assignment

Criterion

- 1.1 CAPACITY for SELF-DIRECTED LEARNING to understand the design principles of software development
- 1.2 ABILITY to EXPLAIN AND APPLY the object-oriented design techniques

Excellent (A+, A, A-)

High

Good (B+, B, B-)

Significant

Fair (C+, C, C-)

Moderate

Marginal (D)

Basic

Failure (F)

Not even reaching marginal levels

Assessment Task

Mid-Term

Criterion

2.1 CAPACITY for SELF-DIRECTED LEARNING to understand the design principles of software development, software development processes and different design patterns and techniques.

Excellent (A+, A, A-)

High

Good (B+, B, B-)

Significant

Fair (C+, C, C-)

Moderate

Marginal (D)

Basic

Failure (F)

Not even reaching marginal levels

Assessment Task

Group project

Criterion

3.1 ABILITY TO EXPLAIN AND DEMONSTRATE IN DETAIL and with ACCURACY methods of software engineering procedures applied from requirement elicitation to developing software design solutions in a project team.

Excellent (A+, A, A-)

High

Good (B+, B, B-)

Significant

Fair (C+, C, C-)

Moderate

Marginal (D)

Basic

Failure (F)

Not even reaching marginal levels

Assessment Task

Examination

Criterion

4.1 CAPACITY for SELF-DIRECTED LEARNING to understand the design principles of software development, software development processes and different design patterns and techniques.4.2 ABILITY to EXPLAIN AND APPLY the Object Oriented Design Techniques

Excellent (A+, A, A-)

High

Good (B+, B, B-)

Significant

Fair (C+, C, C-)

Moderate

Marginal (D)

Basic

Failure (F)Not even reaching marginal levels

Part III Other Information**Keyword Syllabus**

Software Development Process, Requirement Elicitation and Analysis, Use Case Specifications, Software Design Principles, Software Design Patterns, Object-Oriented Software Design Modelling, UML, Class Diagram, Use-Case Diagram, Sequence Diagram, Semantics of UML diagrams, Professional Ethics.

Syllabus

- Software Development Process
Project scope, process issues, software development life cycle models, professional ethics.
- Software Requirements Specification
Requirements elicitation, analysis, use-case modelling, specification and documentation.
- Object-Oriented Analysis (OOA)
Object-oriented concepts: object modelling, reuse, object interactions and responsibilities.
- Object-Oriented Design (OOD)
Fundamental software design principles, concepts and applications of software design patterns.
- UML
Key types of diagram: use case diagram, class diagram, sequence diagram. Semantics and applications of these diagrams.

Reading List**Compulsory Readings**

Title	
1	Nil

Additional Readings

Title	
1	Sommerville I.(2012) Software Engineering. Addison Wesley, 10th edition.
2	Larman C. (2005) Applying UML and Patterns: Introduction to OOA/D and Iterative Development. Pearson Education, Prentice Hall, 3rd edition.
3	Martin R C, and Martin M. Agile (2006). Principles, Patterns, and Practices in C#. Prentice Hall.
4	Bentley L, and Whitten J.(2007) Systems Analysis & Design for the Global Enterprise. McGraw Hill.