CS3347: SOFTWARE ENGINEERING PRINCIPLES AND PRACTICE

Effective Term

Semester A 2022/23

Part I Course Overview

Course Title

Software Engineering Principles and Practice

Subject Code

CS - Computer Science

Course Number

3347

Academic Unit

Computer Science (CS)

College/School

College of Engineering (EG)

Course Duration

One Semester

Credit Units

3

Level

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

Medium of Instruction

English

Medium of Assessment

English

Prerequisites

CS2313 Computer Programming or equivalent

Precursors

Nil

Equivalent Courses

Nil

Exclusive Courses

Nil

Part II Course Details

Abstract

This course aims to introduce the fundamental principles, methods and practice of team-based software development, with a key focus on the object-oriented analysis and design methodology. Students will appreciate key activities in the development of software applications, including project management, requirements specification and design. Students are prepared to participate in a software development team and to optionally pursue further studies in advanced topics in software engineering.

Course Intended Learning Outcomes (CILOs)

	CILOs	Weighting (if app.)	DEC-A1	DEC-A2	DEC-A3
1	Describe the major software development processes and explain their relative merits and limitations.		x		
2	Describe software project management techniques within a team environment.		x		
3	Specify software requirements and conduct object-oriented modelling, analysis and design of software systems.		x	x	x
4	Apply software engineering techniques to appreciate the professional practice of developing quality software.		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.

Teaching and Learning Activities (TLAs)

	TLAs	Brief Description	CILO No.	Hours/week (if applicable)
1	Lecture	Introduce the fundamental principles and practices of software engineering methodologies utilizing examples of real-life software development and case studies.	1, 2, 3, 4	3 hours/week

2	Tutorial	Clarify concepts, reinforce knowledge, work on small problems or exercises, and discuss their solutions with peers and the instructor.	1, 2, 3, 4	1 hour/week
3	Group Project	Students will have to work as a small team on a software project. The project activities may provide opportunities for students to define software requirements and perform software design and documentation.	3, 4	4 hours/week

Assessment Tasks / Activities (ATs)

	ATs	CILO No.	Weighting (%)	Remarks (e.g. Parameter for GenAI use)
1	Group project	3, 4	40	
2	Test	3	10	

Continuous Assessment (%)

50

Examination (%)

50

Examination Duration (Hours)

2

Additional Information for ATs

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

Assessment Rubrics (AR)

Assessment Task

Group project

Criterion

Ability to identify actors and use cases, and draw the use case diagram with relationships among use cases with descriptions.

Excellent (A+, A, A-)

High

Good (B+, B, B-)

Significant

Fair (C+, C, C-)

Moderate

Marginal (D)

Basic

Failure (F)

Below marginal level

Assessment Task

Group project

Criterion

Ability to identify entity classes with reasonable details, and correct class associations shown in the class diagram, with class description.

Excellent (A+, A, A-)

High

Good (B+, B, B-)

Significant

Fair (C+, C, C-)

Moderate

Marginal (D)

Basic

Failure (F)

Below marginal level

Assessment Task

Group project

Criterion

Ability to identify all necessary boundary classes and control classes consistent with the use case scenarios.

Excellent (A+, A, A-)

High

Good (B+, B, B-)

Significant

Fair (C+, C, C-)

Moderate

Marginal (D)

Basic

Failure (F)

Below marginal level

Assessment Task

Group project

Criterion

Ability to draw sequence diagrams showing the object-object interactions for the use cases matching the corresponding use case scenario description; and eventually adding details to the previous class diagrams.

Excellent (A+, A, A-)

High

Good (B+, B, B-)

Significant

Fair (C+, C, C-)

Moderate

Marginal (D)

Basic

Failure (F)

Below marginal level

Assessment Task

Group project

Criterion

Ability to rewrite the user requirements descriptions with clear elaboration on the assumptions made during the design process.

Excellent (A+, A, A-)

High

Good (B+, B, B-)

Significant

Fair (C+, C, C-)

Moderate

Marginal (D)

Basic

Failure (F)

Below marginal level

Assessment Task

Group project

Criterion

Ability to present all design outputs in the user requirement specification which contains 1/ use case modelling with descriptions, 2/ all categories of classes with descriptions, 3/ sequence diagrams matching use cases, 4/ one state chart diagram.

Excellent (A+, A, A-)

High

Good (B+, B, B-)
Significant

Fair (C+, C, C-)
Moderate

Marginal (D)
Basic

Failure (F)
Below marginal level

Assessment Task
Group project

Criterion
Ability to organize the specification nicely with minimum ambiguities.

Excellent (A+, A, A-)
High

Good (B+, B, B-)
Significant

Fair (C+, C, C-)
Moderate

Marginal (D)
Basic

Failure (F)
Below marginal level

Assessment Task
Test

Criterion
Ability to apply appropriate techniques to model simple user requirements.

Excellent (A+, A, A-)
High

Good (B+, B, B-)
Significant

Fair (C+, C, C-)
Moderate

Marginal (D)
Basic

Failure (F)
Below marginal level

Assessment Task
Examination

Criterion
Ability to achieve the respective CILOs.

Excellent (A+, A, A-)
High

Good (B+, B, B-)
Significant

Fair (C+, C, C-)
Moderate

Marginal (D)
Basic

Failure (F)
Below marginal level

Part III Other Information

Keyword Syllabus

Software process. Software development life cycle models. Software project management. Project scoping, planning and scheduling. Team organization. Documentation. Software specification. Requirements elicitation and analysis. Use case modelling. Object-oriented analysis and design. Object-oriented concepts. Object modelling. Software design principles. Design for change. UML diagrams: use case, activity, class, and sequence diagrams. CASE tools. Software implementation and testing. Integrated software engineering environments. Prototyping. Software quality.

Reading List

Compulsory Readings

	Title
1	Bruegge and Dutoit (2010). Object-Oriented Software Engineering. Pearson. 3rd edition.

Additional Readings

	Title
1	Sommerville I. (2010). Software Engineering (9e). Addison Wesley.
2	Stumpf and Teague (2005). Object-Oriented Systems Analysis and Design with UML. Pearson.
3	Larman C. (2005). Applying UML and Patterns: Introduction to OOA/D and Iterative Development (3e). Pearson Education, Prentice Hall.