fit@hcmus

VNUHCM-UNIVERSITY OF SCIENCE
FACULTY OF INFORMATION TECHNOLOGY

COURSE SYLLABUS

CSC13010 – Software Design

1. GENERAL INFORMATION

Course name: Software Design	Software Design
Course name (in Vietnamese):	Thiết Kế Phần Mềm
Course ID:	CSC13010
Knowledge block:	Specialization
Number of credits:	4
Credit hours for theory:	45
Credit hours for practice:	30
Credit hours for self-study:	90
Prerequisite:	
Prior-course:	Introduction to software engineering
Instructors:	

2. COURSE DESCRIPTION

- ❖ This course aims to equip students with basic skills to analyze and design software.
- ❖ Upon completion this course, students can:
  - ✓ Describe the common principles to analyze and design software from software requirements
  - ✓ Apply object oriented methods and techniques to analyze and design software
  - ✓ Recognize, analyze and evaluate basic pros and cons of an existing analysis or design model, the architecture of a software system, the communication between components in a given system.
  - ✓ Apply basic object oriented techniques to optimize analysis/design models to enhance the evolution and flexibility of software systems.

- This course introduces the common principles to analyze and design software from software requirements.
- The content of this course focuses on object oriented techniques (using UML) to analyze, to design architecture, interface, business logic, and data.
- Several advanced topics can be optionally introduced (e.g. design patterns, service oriented architecture…)

3. COURSE GOALS

At the end of the course, students are able to

ID	Description	Program LOs
G1	Participate actively in group discussions (large groups); divide the work and coordinate work according to plans in a small group (including 2-3 students); be aware of the roles and responsibilities of team members.	2.1.1, 2.1.2, 2.1.3, 2.2.2, 3.1.1, 3.3.2
G2	Reading technical documents in English in the design of software and technologies used; explain some English terms in software design; presentation (in the form of a written report in the prescribed template) and a presentation on the group's topic.	2.3.1, 2.3.2, 2.4.3, 2.4.5
G3	Have critical thinking and holistic thinking when designing software.	2.1.4, 2.1.5
G4	Present and explain the role of software design in the software development process, some of the main approaches to software design.	1.4, 5.2.1, 5.2.2, 5.2.3
G5	Apply object-oriented analysis method and fundamental principles in analysis to analyze software in small and medium scale, towards software evolution.	1.4, 4.1.2, 5.1.3, 5.2.1, 5.2.2, 5.2.3

G6	Apply object-oriented design methods and fundamental principles in the design division to design software in small and medium scale towards software evolution.	1.4, 4.1.2, 5.1.3, 5.2.1, 5.2.2, 5.2.3
G7	Initially assessing the quality of the analysis and design modelling and applying a number of designs to improve the quality of the analysis and design modelling.	1.4, 4.1.1, 4.1.4, 4.3.1, 4.3.2, 4.3.4
G8	Select and use several software engineering tools and environments to analyze and design software at a small and medium scale.	5.3.1, 5.3.2, 5.3.3

4. COURSE OUTCOMES

CO	Description	I/T/U
G1.1	Practice working in groups of 2-4 students, focusing on the software design phase.	U
G1.2	Express correct attitudes, views, and awareness of the role of software systems analysts and designers (Analyst, Designer)	I
G1.3	Practice constantly learning, self-learning, and applying new tools and technologies in software design.	I
G2.1	Summarize technical English documents on software design and analysis	U
G2.2	Use and interpret English terminology for software design and analysis	T
G2.3	Practice presenting the results of software design and analysis in a group and before class	U
G2.4	Write software design and analysis report using given template in Vietnamese	TU
G3.1	Apply critical thinking and holistic thinking when analyzing and designing software	TU

G4.1	Demonstrate and explain the role of software analysis and design in the software development process.	T
G4.2	Distinguish the analyst and designer's responsibilities and work from other roles in the project.	T
G5.1	Practice modeling software requirements for small and medium sized projects with use-case diagrams, activity diagrams, and use- case specifications.	TU
G5.2	Conduct object-oriented software analysis using class diagrams.	TU
G5.3	Conduct object-oriented software analysis using state diagrams, sequence diagrams, communication diagrams	TU
G6.1	Apply basic principles to map analysis model to design model.	TU
G6.2	Design software architecture which consists of multiple components/layer/tier for small and medium-scale software.	TU
G6.3	Design database for software.	TU
G6.4	Apply basic principles of interface to design user interface.	TU
G7.1	Apply basic principles of interface to design user interface.	T
G7.2	Apply object-oriented techniques at a basic level to increase software evolution.	U
G8.1	Select and use proficiently a number of tools and environments suitable for analyzing and designing a particular software application.	IU

5. TEACHING PLAN

THEORY

ID	Topic	Course outcomes	Teaching/Learning Activities (samples)	Assessments
1	Overview and Revision : • Concepts in Software Engineering • Best Practices in Software Development • Concepts in Object-Oriented Programming	G4.1, G2.1, G2.2, G4.2, G1.2, G1.3	Lecturing Q&A	HW#1
2	User Requirement • Gather information and specify the requirement • Identify and collect requirements	G2.4, G1.1, G2.3	Lecturing , Q&A	PR J#1 PR J#5
3	Requirement Modeling • Problem Statement • Actors and Use-cases • Use-case Model • Use-case Diagram • Relationships in Use-case Diagrams: generalization, <<include>>, <<extend>> • Use-case Specification: Scenarios, Flows of Events, Alternatives, Pre-conditions, Post-conditions… • Glossary and Supplemental Specification • Examples	G5.1, G2.4, G1.1, G2.1, G2.2, G2.3, G3.1	Lecturing Demonstration, Discussion	H W# 2 PR J#2 PR J#5

	Requirement Modeling with UML – Activity Diagrams			
4	Class Diagrams • Revisions: basic concepts in OOP and notations in UML • How to enhance the capability for software evolution? State Machine diagrams • Concepts and notations • Examples Analysis Classes • Sequence Diagrams, Communication Diagrams, and VOPCs • Concepts and notations • Examples	G5.2, G5.3, G7.1, G8.1, G2.4, G1.1, G2.3, G3.1	Lecturing, Demonstration, Discussion	HW#5 PRJ#3 PRJ#5
5	Relational Database Design • Mapping from a class diagram to a relational database • How to enhance the capability for software evolution? XML and Semi-structured Data • Introduction to XML • How to store data using XML • Comparison between relational databases and XML-based data	G6.1, G6.3, G7.1, G8.1, G2.4, G1.1, G2.3, G3.1	Lecturing, Demonstration	HW#6 PR J#4 PR J#5

6	Software Architecture • Introduction to Software Architecture • Layers and Tiers • Some guidelines for Software Architecture Analysis and Design	G6.2, G7.2, G2.4, G1.1, G2.3, G3.1	Question & answer Case study & discussion	H W# 7 H W# 8 PR J#5
7	(User) Interface Design • Introduction • Layout and behavior of a (user) interface • Some common approaches for designing user interfaces • Some (common and easy-to-understand) notations • Examples and applications • Data Input Forms: simple object, complex object, relation • Search Forms • Processing Business Forms • Reports • Several Techniques to Enhance Qualities of User Interfaces • Supplemental Information • Supplemental Operations • Action Acceleration • Exception Handlers	G6.4, G2.4, G1.1, G2.3, G3.1	Lecturing Demonstration	H W# 3 H W# 4 PR J#5

8	Miscellaneous • Design Patterns • Software Refactoring	G7.2, G2.4, G1.1,	Lecturing Q&A, Discussion	PRJ#5

	• Late-binding functions • Web services (SOAP, REST) and Service Oriented Architecture • Model-Driven Architecture • Mashups and Widgets	G2.1, G2.2, G2.3, G3.1		

LABORATORY

ID	Topic	Course outcomes	Teaching/Learning Activities (samples)	Assessments
1	ADO.NET	G6.3	Explain and demonstrate Q&A, Group discussion	HW#1
2	User Control Graphic and Sprite (2D)	G6.4	Lecturing Demonstration, Q&A	HW#3 HW#4
3	Linq và XML	G6.3, G1.1, G2.3	Provide instructions Demonstration, discussion	PRJ#5
4	Web Service	G6.2, G1.1, G2.3	Lecturing Demonstration, discussion	PRJ#5
5	MVC	G6.2, G1.1, G2.3	Lecturing Demonstration	PRJ#5

6	Plugin	G6.2, G7.2, G1.1, G2.3	Q&A Case study and discussion	PRJ#5

6. ASSESSMENTS

ID	Topic	Description	Course outcomes	Ratio (%)
HW	**Homework**			**10%**
HW	Homework: HW1, HW2, HW3, HW4, HW5, HW6, HW7, HW8	HW1: ADO.net HW2: Use-Case Diagram HW3: User Control HW4: Graphical Programming HW5: Class Diagram HW6: Database Design HW7 : 3 Layer HW8: 3 Tier	G5.1, G2.4, G1.1, G2.1, G2.3	1.25%
PRJ	**Projects**			**30%**
PRJ#1	System Requirements	Using template file to write	G2.4, G1.1, G2.3	2.5%
PRJ#2	Use Case Diagram	Using template file to write	G5.1, G2.4, G1.1, G2.3	2.5%
PRJ#3	Class Diagram	Using template file to write	G5.2, G5.3, G8.1, G2.4, G1.1, G2.3	2.5%

PRJ#4	Database Design	Using template file to write	G6.1, G6.3, G8.1, G2.4, G1.1, G2.3	2.5%
PRJ#5	Final Project	App, Video, Help File	G5.1, G5.2, G6.1, G6.2, G6.3, G6.4, G7.2, G8.1, G2.4, G1.1, G2.1, G2.2, G2.3, G4.2, G1.2, G1.3	20%
EX	**Exams**			**50%**
EX#1	Midterm exam	Closed book exam. Describe the understanding of topics, analyze & program to solve problems		15%
EX#2	Final exam	Closed book exam. Describe the understanding of different topics, analyze & program to solve problems		35%

7. RESOURCES

Textbooks

*Ian Sommerville, **Software Engineering: (Update) (8th Edition)**, Addison Wesley, 2006.*

***Software Engineering: A Practitioner's Approach**, Roger S Pressman, Roger Pressman, McGraw-Hill Science/Engineering/Math, 2004.*

*Ian Sommerville, **Software Engineering: (Update) (8th Edition)**, Addison Wesley, 2006.*

***Software Engineering - Object-Oriented System Development**, Dennis de Champeaux, Douglas Lea, Penelope, Faure, Addison Wesley, 1993.*

***Software Architecture in Practice, 2nd edition**, Len Bass, Paul Clements, Rick Kazman, Addison Wesley, 2003.*

***Data Access Patterns: Database Interactions in Object-Oriented Applications**, Clifton Nock, Addison Wesley, 2003.*

***Pattern-Oriented Analysis and Design: Composing Patterns to Design Software Systems**, Sherif M. Yacoub, Hany H. Ammar, Addison Wesley, 2003.*

***Enterprise Solution Patterns Using Microsoft .NET**, Microsoft Corporation, Microsoft Press, 2003.*

***Patterns of Enterprise Application Architecture**, Martin Fowler, David Rice, Matthew Foemmel, Edward Hieatt, Robert Mee, Randy Stafford, Addison Wesley, 2002.*

***Design Patterns Explained – A New Perspective on Object Oriented Design**, Alan Shalloway, James R. Trott, Addison Wesley, 2004.*

***Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions**, Gregor Hohpe, Bobby Woolf, Addison Wesley, 2003.*

Others

Visual Studio .NET

Eclipse/Netbean

Microsoft SQL

Rational Rose, Visio, StarUML…

8. GENERAL REGULATIONS & POLICIES

- All students are responsible for reading and following strictly the regulations and policies of the school and university.
- Students who are absent for more than 3 theory sessions are not allowed to take the exams.

- For any kind of cheating and plagiarism, students will be graded 0 for the course. The incident is then submitted to the school and university for further review.
- Students are encouraged to form study groups to discuss on the topics. However, individual work must be done and submitted on your own.