IT4490 THIẾT KẾ VÀ XÂY DỰNG PHẦN MỀM

SOFTWARE DESIGN AND CONSTRUCTION

Phiên bản: 2022.05.20

1. THÔNG TIN CHUNG - GENERAL INFORMATION

Tên học phần:	Thiết kế và xây dựng phần mềm
Coursename:	*Software Design and Construction*
Mã số học phần:	IT4490
Course code:	
Khối lượng:	3(2-2-0-6)
Credit:	- Lý thuyết (Theory): 30 tiết - Bài tập (Exercise): 30 tiết - Thực hành (Experiment): 0 tiết (Có Bài tập lớn - With capstone-project)
Học phần tiên quyết:	Không
Pre-requisite courses:	
Học phần học trước:	IT3180: Nhập môn Công nghệ phần mềm
Prior courses	(IT3180: Introduction to Software Engineering) IT3120: Phân tích thiết kế hệ thống (IT3120: Systems Analysis and Design)
Học phần song hành:	Không
Co-requisite courses:	None

2. MÔ TẢ HỌC PHẦN - COURSE DESCRIPTION

Học phần này cung cấp cho sinh viên kiến thức và kinh nghiệm thực tiễn về phát triển phần mềm theo chuẩn ITSS. Sinh viên cần nắm và vận dụng được về phương pháp phân tích thiết kế hướng đối tượng, các nguyên lý thiết kế, cách thức xây dựng phần mềm có chất lượng tốt, đảm bảo sản phẩm phần mềm có sự móc nối thấp/lỏng lẻo (low/loose coupling) và tính kết dính cao/chặt (high/tight cohesion). Sinh viên được giảng dạy, trao đổi và thực hành với các nguyên lý thiết kế phần mềm S.O.L.I.D nhằm tạo ra các thiết kế tốt, linh hoạt, dễ mở rộng với case study xuyên suốt học phần theo nhóm. Học phần chỉ ra vị trí và vai trò của thiết kế và xây dựng phần mềm trong bức tranh chung trong các môn học liên quan tới quy trình phát triển phần mềm hay phương pháp phân tích thiết kế hướng đối tượng. Các kỹ thuật, phương pháp và công cụ thiết kế kiến trúc và thiết kế chi tiết lần lượt được trình bày cho sinh viên. Đồng thời, sinh viên cũng được chia sẻ về cách thức tái cấu trúc (refactor với Eclipse) nhằm cải tiến chất lượng thiết kế và mã nguồn. Sinh viên được tiếp cận với kỹ thuật phát triển phần mềm hướng kiểm thử và cách viết mã nguồn cho các trường hợp kiểm thử (với JUnit). Sinh viên cũng được giới thiệu tổng quan về các kinh nghiệm thực tế được các kỹ sư lành nghề đúc rút thành các mẫu thiết kế - nội dung sẽ được đi sâu trong chương trình Thạc sĩ kỹ thuật hoặc Kỹ sư.

The course provides students with knowledge and experiences on developing software in compliance with the ITSS industry standard. Students are able to grasp and apply object-

oriented analysis and design, design principles, and construction practices to build a good software with loose coupling and tight cohesion. The students will learn, discuss, present and practice S.O.L.I.D principles with a case study in their teams. The course also shows the role of software construction and design with other courses related to software development process and object-oriented analysis and design methodology. Methods, techniques and tools of the following tasks will be covered: architectural design and detail design, code refactoring (with Eclipse), test-driven software development (with JUnit). Students are also given the overview of design patterns, some of the best practices adapted by experienced object-oriented software developers, which will be deeply studied in the engineer or master program.

3. MỤC TIÊU VÀ CHUẨN ĐẦU RA CỦA HỌC PHẦN - GOAL AND OUTPUT REQUIREMENT

Sinh viên hoàn thành học phần này có khả năng:

After completing the course, students are able to:

Mục tiêu/ Course learning outcomes	**Mô tả mục tiêu/Chuẩn đầu ra của học phần** **Description of course learning outcomes**	**CĐR được phân bổ cho HP/ Mức độ (I/T/U)** **Mapping to Program learning outcomes (I/T/U)**
[1]	**[2]**	**[3]**
M1	**Hiểu và biết cách vận dụng quy trình thiết kế và xây dựng phần mềm vào một bài toán thực tế** **Understand and be able to apply the software design and construction process to a real problem**	1.2.6; 1.3.1; 2.1.1÷2.1.4; 2.3.1÷2.3.4; 4.4.1; 4.4.2
M1.1	Hiểu và biết cách vận dụng đúng quy trình thiết kế và xây dựng phần mềm một cách hợp lý vào một bài toán thực tế Understand and be able to properly apply the software design and construction process to a real problem in a reasonable manner	[1.2.6] (U) [1.3.1] (TU) [2.1.1÷2.1.4] (U) [2.3.1÷2.3.4] (U) [4.4.1, 4.4.2] (U)
M1.2	Hiểu và biết quy trình tối ưu thiết kế từ thiết kế sơ bộ sang thiết kế chi tiết cho một bài toán thực tế Understand the process to optimize the design, from the preliminary design to the detailed design for a real problem	[1.2.6] (U) [1.3.1] (TU) [2.1.1÷2.1.4] (U) [2.3.1÷2.3.4] (U) [4.4.1, 4.4.2] (U)
M1.3	Hiểu và biết cách hiện thực hoá kết quả thiết kế sang mã nguồn và kiểm thử đơn vị Understand the process to transform the design into source code and unit testing	[1.2.6] (U) [1.3.1] (TU)
M2	**Hiểu và biết cách sử dụng các mức độ của các tính chất phần mềm và các kỹ thuật hướng đối tượng nhằm đánh giá chất lượng của bản thiết kế và tạo ra**	1.3.1; 2.1.1÷2.1.4; 2.3.1÷2.3.4

	những bản thiết kế tốt **Understand the quality properties of software and the values of object-oriented techniques to evaluate the quality of a design, be able to create good designs**	
M2.1	Hiểu và áp dụng đúng và hiệu quả các kỹ thuật hướng đối tượng Kế thừa và Đa hình trong thiết kế phần mềm Understand and be able to apply inheritance and polymorphism principles in software design	[1.2.1] (U) [1.3.1] (TU)
M2.2	Hiểu, biết cách xác định và lựa chọn các mức độ của tính kết nối Coupling (Content, Common, External, Control, Stamp, Data, Message) nhằm tạo ra bản thiết kế tốt. Understand and be able to identify and choose the types of coupling (Content, Common, External, Control, Stamp, Data, Message) to create a good design	[1.2.1] (U) [1.3.1] (TU)
M2.3	Hiểu, biết cách xác định và lựa chọn các mức độ của tính kết dính Cohesion (Coincidental, Logical, Temporal, Procedural, Communicational, Sequential, Functional) nhằm tạo ra bản thiết kế tốt. Understand and be able to identify and choose the types of cohesion (Content, Common, External, Control, Stamp, Data, Message) to create a good design	[1.2.1] (U) [1.3.1] (TU)
M2.4	Hiểu và biết cách sử dụng các kiến thức về tính móc nối (cần lỏng lẻo) và tính kết dính (cần chặt) để đánh giá chất lượng của bản thiết kế Understand how to use the loose coupling and high cohesion principles to evaluate the quality of design	[1.2.1] (U) [1.3.1] (TU) [2.1.1÷2.1.4] (U) [2.3.1÷2.3.4] (U)
M3	**Hiểu và biết vận dụng các kiến thức về nguyên lý thiết kế S.O.L.I.D vào thiết kế hướng đối tượng hiệu quả và tối ưu bản thiết kế, ứng dụng trong bài tập lớn** **Understand and be able to apply S.O.L.I.D design principles to optimize the design**	1.2.1; 1.3.1; 2.1.1÷2.1.4; 2.3.1÷2.3.4
M3.1	Hiểu và nắm được vai trò của các kinh nghiệm thực tiễn cũng như các nguyên lý thiết kế trong việc tạo ra các thiết kế tốt, dễ mở rộng và bảo trì Understand the best practices and design principles to create good, scalable, and maintainable designs	[1.2.1] (U) [1.3.1] (TU) [2.1.1÷2.1.4] (U)
M3.2	Hiểu và biết cách sử dụng năm (05) nguyên lý S.O.L.I.D nhằm tạo ra được những thiết kế tốt đồng thời biết cách tối ưu thiết kế phần mềm Understand and be able to apply 5 S.O.L.I.D principles to create and optimize the design	[1.2.1] (U) [1.3.1] (TU) [2.1.1÷2.1.4] (U) [2.3.1÷2.3.4] (U)
M4	**Sử dụng thành thạo một ngôn ngữ lập trình hướng đối tượng để hiện thực hóa các nguyên lý thiết kế** **Use an object-oriented programming language to implement the design**	1.2.1; 1.3.1; 2.1.1÷2.1.4; 2.3.1÷2.3.4
M4.1	Lập trình theo quy ước viết mã nguồn và phong cách lập	[1.2.1] (U)

	trình thống nhất Be able to follow the coding conventions to create high-quality source code	[1.3.1] (TU)
M4.2	Biết cách tái cấu trúc nhằm cải thiện mã nguồn Apply refactoring techniques to improve the source code	[1.2.1] (U) [1.3.1] (TU) [2.1.1÷2.1.4] (U) [2.3.1÷2.3.4] (U)
M4.3	Hiểu và biết cách hiện thực các thiết kế theo các nguyên lý thiết kế bằng một ngôn ngữ lập trình hướng đối tượng phổ biến trong thực tế Be able to implement designs, using a popular object-oriented programming language	[1.2.1] (U) [1.3.1] (TU)
M5	**Hiểu và vận dụng các kỹ thuật kiểm thử vào kiểm thử đơn vị cho các đơn vị phần mềm** **Understand and be able to perform unit tests**	1.2.1; 1.3.1
M5.1	Hiểu về phát triển phần mềm hướng kiểm thử Understand Test-Driven Development (TDD)	[1.2.6] (U) [1.3.1] (TU)
M5.2	Biết cách viết các trường hợp kiểm thử sử dụng thư viện kiểm thử có sẵn như JUnit Be able to write test cases, using a unit testing framework, e.g., JUnit	[1.2.1] (U) [1.3.1] (TU)
M6	**Có kỹ năng xã hội cần thiết để làm việc hiệu quả trong nhóm đa ngành và trong môi trường quốc tế** **Have the social skills to work effectively in a multidisciplinary team and in an international environment**	3.1.1; 3.1.2; 3.1.3; 3.1.4
M6.1	Chủ động tham gia cũng như có khả năng thành lập nhóm phù hợp với công việc Be able to work actively in a team, and form a new team	[3.1.1] (IU)
M6.2	Tổ chức các hoạt động nhóm Be able to organize group activities	[3.1.2] (IU)
M6.3	Quản lý tiến trình hoạt động của nhóm Be able to manage group activities	[3.1.3] (IU)
M6.4	Có khả năng hợp tác, phối hợp với các thành viên khác trong nhóm, giải quyết vấn đề Be able to cooperate and coordinate with other team members	[3.1.4] (IU)
M6.5	Có kỹ năng thuyết trình Have presentation skills	3.2.2 (U)

4. TÀI LIỆU HỌC TẬP - REFERENCES

Giáo trình

Textbooks

[1] Robert C. Martin (2014). *Agile Software Development, Principles, Patterns, and Practices*. Pearson Higher Education.

[2] Roger S. Pressman (2014). *Software Engineering – A Practitioner Approach (8th Edition)*. McGaw Hill Education.

Sách tham khảo

Reference books

[1] IEEE 12207-2017 - *ISO/IEC/IEEE International Standard - Systems and software engineering -- Software life cycle processes*. IEEE Standards Association.

[2] James Rumbaugh, Ivar Jacobson, Grady Booch (2005). *The Unified Modeling Language Reference Manual (2nd Edition)*. Addison-Wesley Professional.

[3] Ivar Jacobson (1992). *Object-Oriented Software Engineering: A Use Case Driven Approach*. Addison-Wesley Professional.

[4] Marian Petre, André van der Hoek (2016). *Software Design Decoded: 66 Ways Experts Think*. The MIT Press.

[5] Robert C. Martin (2008). *Clean code: a handbook of agile software craftsmanship / Robert C. Martins*. Prentice Hall.

5. CÁCH ĐÁNH GIÁ HỌC PHẦN - EVALUATION

Điểm thành phần **Module**	**Phương pháp đánh giá cụ thể** **Evaluation method**	**Mô tả** **Detail**	**CĐR được đánh giá** **Output**	**Tỷ trọng** **Percent**
[1]	**[2]**	**[3]**	**[4]**	**[5]**
A1. Điểm quá trình **Mid-term (*)**	**Đánh giá quá trình** **Progress**			**50%**
	A1.1. Thảo luận trên lớp Discussion	Chuyên cần Diligence Thuyết trình Presentation		10%
	A1.2. Bài tập về nhà Homework	Thiết kế Design Lập trình Programming		20%
	A1.3. Bài tập nhóm Capstone Project	Báo cáo Bảo vệ tại lớp Presentation		20%
A2. Điểm cuối kỳ **Final term**	**A2.1. Thi cuối kỳ** **Final exam**	Thi viết Written exam		**50%**

** Điểm quá trình sẽ được điều chỉnh bằng cách cộng thêm điểm chuyên cần. Điểm chuyên cần có giá trị từ –2 đến +1, theo Quy chế Đào tạo đại học hệ chính quy của Trường ĐH Bách khoa Hà Nội.*

The evaluation about the progress can be adjusted with some bonus. The bonus should belong to [-2, +1], according to the policy of Hanoi University of Science and Technology.

6. KẾ HOẠCH GIẢNG DẠY - SCHEDULE

Tuần Week	Nội dung Content	CĐR học phần Output	Hoạt động dạy và học Teaching activities	Bài đánh giá Evaluated in
[1]	[2]	[3]	[4]	[5]
1	**Tổng quan về Thiết kế và xây dựng phần mềm** 1.1 Quy trình phát triển phần mềm *Nhắc lại theo chuẩn quốc tế IEEE 12207:2017 ISO/IEC về Quy trình vòng đời phần mềm* 1.2 Vị trí, vai trò của Thiết kế và xây dựng phần mềm *Kết nối với các môn học khác từ đó nhấn mạnh vào trọng tâm của môn học để sinh viên thấy được điểm khác biệt và điểm kết nối với các môn khác.* 1.3 Quy trình Thiết kế và xây dựng phần mềm *Bao gồm Thiết kế kiến trúc, thiết kế chi tiết, lập trình và kiểm thử đơn vị.* 1.4 Các hoạt động chính trong Thiết kế và xây dựng phần mềm 1.5 Thiết kế phần mềm hướng đối tượng 1.6. Thiết kế cấu trúc và thiết kế hành vi Thiết kế các khía cạnh tĩnh (cấu trúc) và động (hành vi) của phần mềm 1.7 Bài tập lớn: Giao đề bài, thảo luận **Introduction to Software design and construction** 1.1 Software development process 1.2 Position and role of software design and construction 1.3 Software design and construction process 1.4 Activities in software design and construction process 1.5 Design object-oriented software 1.6 Structural design and behavioral	M1.1 M1.2 M6.1 M6.2 M6.3 M6.4	Giảng bài (2T) **<u>Bài tập (2T):</u>** Sinh viên thành lập nhóm, thảo luận về đề bài, hướng dẫn và yêu cầu do giáo viên cung cấp. Teaching **<u>Exercising:</u>** Group formation and discussion	A2.1

	design 1.7 Assignment			
2	**Thiết kế kiến trúc** 2.1 Mục đích của Thiết kế kiến trúc 2.2 Các hoạt động chính của Thiết kế kiến trúc 2.3. Các kiến trúc phần mềm điển hình Mẫu kiến trúc n-tầng, Mẫu kiến trúc MVC, Mẫu kiến trúc pipes-and-filters 2.4 Tìm các thành phần phần mềm *Lý thuyết + Bài tập trên Case Study* 2.5 Thiết kế lớp phân tích *Lý thuyết + Bài tập trên Case Study* **Software Architectural Design** 2.1 Goals of architectural design 2.2. Activities in architectural design 2.3. Classical software architectures 2.4 Identify software components 2.5 Analysis classes	M1.1 M1.2	Đọc trước tài liệu; Giảng bài (2 T) **Bài tập (2 T):** Sinh viên lựa chọn kiến trúc phần mềm và tìm các thành phần mềm theo kiến trúc đã lựa chọn Note reading Teaching **Exercising:** Students choose software architecture and find software components according to the selected architecture	A1.1 A1.3 A2.1
3	**Thiết kế kiến trúc (tiếp)** 2.6 Thiết kế tương tác giữa các thành phần phần mềm *Lý thuyết + Bài tập trên Case Study* 2.7 Thiết kế lớp phân tích *Lý thuyết + Bài tập trên Case Study* **Softwaral Architecture Design (cont.)** 2.6 Software connector 2.7 Analysis classes	M1.1 M1.2	Đọc trước tài liệu; Giảng bài (2 T) **Bài tập (2 T):** Thiết kế kiến trúc: Sinh viên thiết kế tương tác giữa các thành phần tìm thấy trong bài trước trên Bài tập lớn, vẽ lớp phân tích Note reading Teaching **Exercising:** Architectural design:	A1.1 A1.3 A2.1

			Students design interactions between components found in the previous lesson, draw analysis class	
4	**Thiết kế chi tiết** 3.1 Mục đích của Thiết kế chi tiết 3.2 Các hoạt động chính của Thiết kế chi tiết 3.3 Thiết kế lớp *Lý thuyết + Bài tập trên Case Study* **Detailed design** 3.1 Goals of detailed design 3.2 Activities in detailed design 3.3 Design classes	M1.1 M1.2 M6.1 M6.2 M6.3 M6.4	Đọc trước tài liệu; Giảng bài (2 T) **Bài tập (2 T):** Thảo luận nhóm, trao đổi để đưa ra Thiết kế lớp cho Case Study Note reading Teaching **Exercising:** Group discussion: design class	A1.1 A1.3 A2.1
5	**Thiết kế chi tiết (tiếp)** 3.4 Thiết kế giao diện 3.4.1 Thiết kế giao diện người dùng *Lý thuyết + Bài tập trên Case Study* 3.4.2 Thiết kế giao diện với hệ thống khác *Lý thuyết + Bài tập trên Case Study* **Detailed design (cont.)** 3.4 Interface design 3.4.1 GUI design 3.4.2 System/Device interface design	M1.1 M1.2 M6.1 M6.2 M6.3 M6.4	Đọc trước tài liệu; Giảng bài (2 T) **Bài tập (2 T):** Thiết kế lớp, Thiết kế giao diện với hệ thống khác: Thảo luận nhóm, trao đổi để đưa ra Thiết kế giao diện cho Case Study Note reading Teaching **Exercising:** Class design, interface design. Group	A1.1 A1.3 A2.1

			discussion: interface design	
6	**Thiết kế chi tiết (tiếp)** 3.5 Mô hình hoá dữ liệu 3.5.1 Biểu đồ thực thể liên kết 3.5.2 Các loại mô hình dữ liệu *Mô hình phân cấp, mô hình mạng, mô hình quan hệ, mô hình phi quan hệ* 3.5.3 Thiết kế CSDL *Lý thuyết + Bài tập trên Case Study* **Detailed design (cont.)** 3.5 Data modeling 3.5.1 Entity relationship diagram 3.5.2. Logical Data Model 3.5.3 Database design	M1.1 M1.2 M6.1 M6.2 M6.3 M6.4	Đọc trước tài liệu; Giảng bài (2 T) **Bài tập (2 T):** Thảo luận nhóm, trao đổi để đưa ra Mô hình hoá dữ liệu cho Case Study Note reading Teaching **Exercising:** Group discussion: data modeling	A1.1 A1.3 A2.1
7	**Kiểm thử đơn vị** 4.1 Vai trò của kiểm thử đơn vị 4.2 Thiết kế trường hợp kiểm thử 4.3 Phát triển phần mềm hướng kiểm thử (TDD) 4.4 Kiểm thử đơn vị với JUnit *Bài tập trên Case Study* **Unit Testing** 4.1 Overview 4.2 Test case design 4.3 Test-driven software development 4.4 JUnit testing framework	M5.1 M5.2	Đọc trước tài liệu; Giảng bài (2 T) **Bài tập (2 T):** Thiết kế giao diện người dùng, Mô hình hoá dữ liệu. Sinh viên thiết kế các trường hợp kiểm thử cho Case Study Note reading Teaching **Exercising:** Interface design and data modeling Design test cases	A1.1 A1.3 A2.1
8	**Lập trình** 5.1 Mã nguồn tốt 5.2 Chuẩn lập trình	M1.3 M4.1 M4.2 M4.3	Đọc trước tài liệu; Giảng bài (2 T);	A1.1 A1.3 A2.1

	5.3 Tái cấu trúc mã nguồn 5.4 Tổ chức mã nguồn 5.5 Framework lập trình 5.6 Tích hợp và quản lý mã nguồn *Làm việc nhóm với công cụ quản lý mã nguồn Git: Làm việc nhóm trên nhánh cá nhân và nhánh chung, Tích hợp nguồn, Xử lý đụng độ* *Bài tập trên Case Study* **Programming** 5.1 High-quality source code 5.2 Coding Standards 5.3 Refactoring 5.4 Organize source code 5.5 Programming frameworks 5.6 Version control system	M6.1 M6.2 M6.3 M6.4	**Bài tập (2 T):** Sinh viên thảo luận nhóm về cách tổ chức mã nguồn, framework lập trình và quy ước viết mã nguồn chung cho nhóm. Tiến hành lập trình. Note reading Teaching **Exercising:** Group discussion: organizing source code, coding convention	
9	**Các khái niệm cơ bản trong thiết kế** 6.1 Các mức thiết kế 6.2 Tính mô-đun hoá 6.3 Tính móc nối và tính kết dính *Coupling và Cohesion* *Thế nào là một thiết kế tốt* 6.4 Tính móc nối 6.4.1 Móc nối nội dung 6.4.2 Móc nối thành phần chung 6.4.3 Móc nối điều khiển 6.4.4 Móc nối dữ liệu phức 6.4.5 Móc nối thông điệp *Bài tập trên Case Study* **Design concepts** 6.1 Design levels 6.2 Modular design 6.3 Coupling and cohesion 6.4 Coupling 6.4.1 Content Coupling 6.4.2 Common Coupling 6.4.3 Control Coupling 6.4.4 Stamp Coupling	M1.1 M1.2 M6.1 M6.2 M6.3 M6.4	Đọc trước tài liệu; Giảng bài (2 T) **Bài tập (2 T):** Kiểm tra bản thiết kế hiện giờ đã phải là bản thiết kế tốt chưa, đã đáp ứng các nguyên lý cơ bản chưa? Viết mã nguồn kiểm thử với các trường hợp kiểm thử đã thiết kế cho mã nguồn đã/sẽ viết. Lập trình một số tính năng chính chú ý tuân thủ quy ước lập trình. Note reading	A1.1 A1.3 A2.1

	6.4.5 Message Coupling		Teaching **Exercising:** Evaluate and improve designs Test cases and write high-quality source code	
10	**Các khái niệm cơ bản trong thiết kế (tiếp)** 6.5 Tính kết dính 6.5.1 Kết dính trùng khớp 6.5.2 Kết dính logic 6.5.3 Kết dính hướng thời gian 6.5.4 Kết dính thủ tục 6.5.5 Kết dính giao tiếp 6.5.6 Kết dính hàm 6.5.6 Functional Cohesion *Bài tập trên Case Study* **Design concepts (cont.)** 6.5 Cohesion 6.5.1 Coincidental Cohesion 6.5.2 Logical Cohesion 6.5.3 Temporal Cohesion 6.5.4 Procedural Cohesion 6.5.5 Communicational Cohesion	M2.1 M2.2	Đọc trước tài liệu; Giảng bài (2 T) **Bài tập (2 T):** So sánh các mức độ móc nối. Thực hiện đánh giá các mức độ này trên Case Study Note reading Teaching **Exercising:** Compare cohesion types	A1.1 A1.3 A2.1
11	**Các nguyên lý thiết kế S.O.L.I.D** 7.1 Giới thiệu về các nguyên lý thiết kế 7.2 Giới thiệu về các nguyên lý S.O.L.I.D 7.3 Nguyên lý một nhiệm vụ 7.4 Nguyên lý đóng mở 7.5 Nguyên lý thay thế Liskov 7.6 Nguyên lý chia tách giao diện 7.7 Nguyên lý đảo ngược sự phụ thuộc *Bài tập trên Case Study* **SOLID principles** 7.1 Introduction to design principles 7.2 Introduction to SOLID principles 7.3 Single responsibility principle 7.4 Open close principle	M3.1 M3.2	Đọc trước tài liệu; Giảng bài (2 T) **Bài tập (2 T):** Các bài tập nhỏ theo từng nguyên lý Kiểm tra thiết kế đã có đã thoả mãn các nguyên lý thiết kế SOLID chưa? Đánh giá bản thiết kế dựa trên các nguyên lý về	A1.1 A1.3 A2.1

	7.5 Liskov substitution principle 7.6 Interface segregation principle 7.7 Dependency inversion principle		móc nối lỏng và kết dính chặt, các nguyên lý SOLID. Từ đó thiết kế lại, tái cấu trúc, chỉnh sửa mã nguồn để có bản thiết kế tốt hơn tuân thủ các nguyên lý nói trên Note reading Teaching **Exercising:** Check that the existing designs meet the SOLID design principles? Evaluate the designs and improve them	
12	**Giới thiệu về mẫu thiết kế** 8.1 Khái niệm 8.2 Vai trò của mẫu thiết kế 8.3 Các thành phần của mẫu thiết kế 8.4 Phân nhóm mẫu thiết kế 8.5 Bài tập về mẫu thiết kế Singleton **Design patterns** 8.1 Definition 8.2 Roles of design patterns 8.3 Elements of a design pattern 8.4 Design pattern categories 8.5 Singleton desgin pattern	M2.1 M3.1 M4.3	Đọc trước tài liệu; Giảng bài (2 T) **Bài tập (2 T):** Thiết kế và lập trình mẫu thiết kế Singleton Note reading Teaching **Exercising:** Singleton design pattern	A1.1 A1.2 A1.3 A2.1
13	**Thảo luận, bảo vệ Bài tập lớn** - Sinh viên thuyết trình và bảo vệ BTL - Giảng viên góp ý, hướng dẫn, định hướng **Discussion and presentation** - Assignment presentation - Discussion	M6.1 M6.2 M6.3 M6.4 M6.5	Thuyết trình; Báo cáo bài tập nhóm; Góp ý; Hướng dẫn; Presentation Comment	A1.1 A1.3 A2.1

			Discussion	
14	**Thảo luận, bảo vệ Bài tập lớn** - Sinh viên thuyết trình và bảo vệ BTL - Giảng viên góp ý, hướng dẫn, định hướng **Discussion and presentation** - Assignment presentation - Discussion	M6.1 M6.2 M6.3 M6.4 M6.5	Thuyết trình; Báo cáo bài tập nhóm; Góp ý; Hướng dẫn; Presentation Comment Discussion	A1.1 A2.1
15	**Thảo luận, bảo vệ Bài tập lớn** - Sinh viên thuyết trình và bảo vệ BTL - Giảng viên góp ý, hướng dẫn, định hướng **Tổng kết và ôn tập** **Discussion and presentation** - Assignment presentation - Discussion **Summary and revision**	M6.1 M6.2 M6.3 M6.4 M6.5	Thuyết trình; Báo cáo bài tập nhóm; Góp ý; Hướng dẫn; Presentation Comment Discussion	A1.1 A2.1

7. QUY ĐỊNH CỦA HỌC PHẦN - COURSE REQUIREMENT

- Chủ động đọc trước tài liệu giáo trình, in bài giảng (*.pdf), chuẩn bị sẵn các câu hỏi.
- Dự lớp đầy đủ, theo dõi ghi chú vào tập bài giảng, chủ động đặt câu hỏi, tích cực tham gia phần thảo luận trên lớp.
- Làm bài tập về nhà đầy đủ theo yêu cầu của giảng viên.
- Thực hành cài đặt và sử dụng các công cụ theo hướng dẫn của giảng viên.
- Hoàn thành đầy đủ các nội dung của bài tập lớn (làm bài tập lớn và thảo luận theo nhóm từ 3-5 người), có báo cáo và bảo vệ tại lớp.

- **Nếu môn học được giảng dạy theo hình thức blended learning:**
  - SV tự học online ở nhà qua hệ thống LMS trước khi đến học tại giảng đường; nội dung tự học online gồm đọc tài liệu, xem bài giảng video, làm bài trắc nghiệm.
  - Buổi học trên lớp sinh viên sẽ thảo luận và thực hiện các bài tập giảng viên giao cho.

- Students should read textbook and lectures, print lectures (.pdf)*
- Students should be required to attend classes.
- Students need to complete exercise and homeworks.
- Complete the capstone project (in groups 3-4 members)

8. NGÀY PHÊ DUYỆT - APPROVAL DATE: ……

Chủ tịch Hội đồng **Committee chair**	**Nhóm xây dựng đề cương** **Course development team**

Nguyễn Thị Thu Trang
Trịnh Tuấn Đạt
Nguyễn Mạnh Tuấn

9. QUÁ TRÌNH CẬP NHẬT - UPDATE PROCESS

STT No.	Nội dung điều chỉnh Content of the update	Ngày tháng được phê duyệt Date accepted	Áp dụng từ kỳ/ khóa Applicable from	Ghi chú Note
1	Thay đổi thiết kế học phần "Thiết kế và xây dựng phần mềm" với cấu trúc 3(2-1-1-6) sang cấu trúc mới 3(2-2-0-6) và một số nội dung trong kế hoạch giảng dạy		20221	