UNIVERSITY OF TECHNOLOGY AND EDUCATION HOCHIMINH CITY
Faculty of Information Technology

Major: Information Technology
Level: Undergraduate
Program: Information Technology

Course Syllabus

1. Vietnamese name: Thiết kế phần mềm hướng đối tượng

2. Course name: Object-Oriented Software Design **Course code: OOSD330879**

3. Credits: 3 (2/1/6) (2 theory credits, 1 practice credit)

4. Lecturers:

1/ Main Lecturer: Nguyễn Trần Thi Văn, MSc.

2/ List of other lecturers:

2.1/ Đặng Thanh Dũng, PhD.

2.2/ Lê Văn Vinh, PhD.

5. Prerequisites: NIL

Previous courses: Object-oriented Programming (OOPR230279)

6. Course description:

This course provides students with fundamental knowledge about software analysis and design using object-oriented techniques, Unified Modeling Language (UML) and UML tools, as well as design patterns and their applications in software development. After the course, students are equipped with analysis and design skills; the ability to use UML tools to create software design components; together with the ability to apply appropriate design patterns in various situations to improve software resilience and extensibility. Moreover, soft skills such as teamwork and public presentation are also focused.

7. Course goals:

Goals	Goal description *This course equips students with:*	Expected Learning Outcome of the Programme	Competency Level
G1	Object-oriented analysis and design skills in software development	ELO4 (2.1) ELO6 (2.3)	3 3
G2	Effective teamwork and oral presentation skills	ELO9 (3.1) ELO10 (3.2)	3 4
G3	Ability to apply various design patterns to different situations and problems in software engineering	ELO14 (4.3) ELO15 (4.4) ELO16 (4.5)	5 5 4

8. Course Learning Outcomes (CLOs):

CLOs		Description *After finishing this course, students are able to:*	Programme's ELOs	Competency Level
G1	G1.1	Apply UML in analyzing and designing software	ELO4 (2.1)	3
	G1.2	Illustrate class diagrams for a specific software	ELO4 (2.1)	3
	G1.3	Apply design patterns for a specific problem in software development	ELO6 (2.3)	3
G2	G2.1	Work effectively in a group	ELO9 (3.1)	3
	G2.2	Perform oral presentation before class using laptops and projectors	ELO10 (3.2)	4
G3	G3.1	Evaluate the pros and cons of a software design	ELO14 (4.3)	5
	G3.2	Modify a model for a software using object-oriented techniques.	ELO15 (4.4)	5
	G3.3	Build design patterns for real-world and coding problems.	ELO16 (4.5)	4

9. Ethics:

- **Being honest** in doing homework and tests. ANY plagiarism will result in a score of zero (0) and auto-disqualification for the final test / project.
- **Being diligent**.
- **Being willing to share** knowledge with others.

10. Implementation plan (15 weeks):

Week	Contents	CLOs	Comp. Level	Teaching Method	Assessment Method
1	***Chapter 1: Overview (4/0/8)***				
	A/ Main contents teaching in class *(4)* + Software development process + Basic terms and definitions in object-oriented analysis and design + Object-oriented methodologies	G1.1 G1.2	3 3	Lecture, Group discussion	Short questions
	***B/* Homework and self-studying tasks**: *(8)*	G1.1 G1.2	3 3	Group work	Portfolios

	+ Read documents and write a summary report about UML models				
2	***Chapter 2: User requirements (4/0/8)***				
	A/ Main contents teaching in class *(4)* **+** Usecase diagrams + Actors + Usecases + Relationships + Exercises	G1.1, G1.2	3 3	Lecture, Group work, Case study	Exercises, Portfolios
	***B/* Homework and self-studying tasks**: *(8)* + Do exercises about identifying usecases for a specific module / software	G1.1 G1.2	3 3	Group work, Case study	Exercises, Portfolios
3	***Lab practice for chapter 1 & 2 (0/4/4)***				
	A/ Main contents teaching in class *(4)* + Introduction to Enterprise Architect (EA) + Using EA to model user requirements	G1.1 G1.2	3 3	Lecture, Lab practice	Exercises, Portfolios
	***B/* Homework and self-studying tasks**: *(4)* + Using Enterprise Architect to create class diagrams + Lab exercise	G1.1 G1.2	3 3	Group work, Case study, Lab practice	Exercises, Portfolios
4	***Chapter 3: Class design (4/0/8)***				
	A/ Main contents teaching in class *(4)* **+** Class stereotypes + Components within a class + Methods to identify classes for a module / usecase + Exercises	G1.1 G1.2	3 3	Lecture, Group work, Case study	Exercises, Portfolios
	***B/* Homework and self-studying tasks**: *(8)* + Do exercises about identifying classes for a specific module / software	G1.1 G1.2	3 3	Group work, Case study, Lab practice	Exercises, Portfolios
5	***Chapter 3: Class design (cont) (4/0/8)***				
	A/ Main contents teaching in class *(4)* + Relationships between classes + Dependency + Association	G1.1 G1.2	3 3	Lecture, Group work, Case study	Exercises, Portfolios

	+ Aggregation + Composition + Inheritance + Realization + Class design exercises				
	***B/* Homework and self-studying tasks**: *(8)* + Do exercises about identifying classes for a specific module / software	G3.1 G3.2	5 5	Group work, Case study, Lab practice	Exercises, Portfolios
6	***Lab practice for chapter 3 (0/4/4)***				
	A/ Main contents teaching in class *(4)* + Using Enterprise Architect to create class diagrams + Lab exercise	G3.1 G3.2 G3.3	5 5 4	Lecture, Lab practice	Exercises, Portfolios
	***B/* Homework and self-studying tasks**: *(4)* + Do lab exercises about class diagrams using Enterprise Architect	G1.1 G1.2	3 3	Group work, Case study, Lab practice	Exercises, Portfolios
7	***Chapter 4: Interaction diagrams (4/0/8)***				
	A/ Main contents teaching in class *(4)* + Sequence diagrams - Definition and usage - Components - How to build a sequence diagram + Exercises about sequence diagrams + Communication diagrams - Definition and usage - Components - How to build a communication diagram + Exercises about communication diagrams	G1.1 G1.2	3 3	Lecture, Group work, Case study	Exercises, Portfolios
	***B/* Homework and self-studying tasks**: *(8)* + Do exercises about building interaction diagrams for a specific usecase	G2.1 G2.2	3 4	Group work, Case study, Lab practice	Exercises, Portfolios
8	***Lab practice for chapter 4 (0/4/4)***				
	A/ Main contents teaching in class *(4)* + Using Enterprise Architect to create sequence and communication diagrams + Lab exercise	G1.1 G1.2	3 3	Lecture, Lab practice	Exercises, Portfolios

	***B/* Homework and self-studying tasks**: *(4)* + Do lab exercises about sequence and communication diagrams using Enterprise Architect	G1.1 G1.2	3 3	Group work, Case study, Lab practice	Exercises, Portfolios
9	***Chapter 5: Software design patterns (4/0/8)***				
	A/ Main contents teaching in class *(4)* + Introduction about software design patterns + History + Design pattern template + Classification + Usage and benefits + Examples	G1.3	3	Presentation, Group work, Case study	Portfolios, Rubrics
	***B/* Homework and self-studying tasks**: *(8)* + Read relevant documents and write a report about groups of design patterns + Prepare a presentation about one of the creational patterns	G1.3	3	Group work,	Exercises, Portfolios
10	***Chapter 5: Software design patterns (cont) (2/2/6)***				
	A/ Main contents teaching in class *(4)* Group presentation about patterns: + Factory method + Abstract Factory + Builder + Prototype + Singleton	G1.3 G2.1 G2.2	3 3 4	Presentation, Group work, Case study	Portfolios, Rubrics
	***B/* Homework and self-studying tasks**: *(6)* + Do lab exercises about creational patterns + Prepare a presentation about one of the structural patterns	G1.3	3	Group work, Case study, Lab practice	Exercises, Portfolios
11	***Chapter 5: Software design patterns (cont) (2/2/6)***				
	A/ Main contents teaching in class *(4)* Group presentation about patterns: + Adapter + Composite	G1.3 G2.1 G2.2	3 3 4	Presentation, Group work, Case study	Portfolios, Rubrics

	+ Decorator + Façade + Bridge				
	***B/* Homework and self-studying tasks**: *(6)* + Do lab exercises about structural patterns + Prepare a presentation about one of the behavioral patterns	G1.3	3	Group work, Case study, Lab practice	Exercises, Portfolios
12	***Chapter 5: Software design patterns (cont) (2/2/6)***				
	A/ Main contents teaching in class *(4)* Group presentation about patterns: + Iterator + Command + Observer + Interpreter + Mediator	G1.3 G2.1 G2.2	3 3 4	Presentation, Group work, Case study	Portfolios, Rubrics
	***B/* Homework and self-studying tasks**: *(6)* + Do lab exercises about learned behavioral patterns + Prepare a presentation about one of the behavioral patterns	G1.3	3	Group work, Case study, Lab practice	Exercises, Portfolios
13	***Chapter 5: Software design patterns (cont) (2/2/6)***				
	A/ Main contents teaching in class *(4)* Group presentation about patterns: + State + Strategy + Template method + Visitor	G1.3 G2.1 G2.2	3 3 4	Presentation, Group work, Case study	Portfolios, Rubrics
	***B/* Homework and self-studying tasks**: *(6)* + Do lab exercises about learned behavioral patterns	G1.3	3	Group work, Case study, Lab practice	Exercises, Portfolios
14	***Chapter 5: Software design patterns (cont) (2/2/6)***				
	A/ Main contents teaching in class *(4)* + Lab exercises about design patterns	G3.1 G3.2	5 5	Lecture, Lab practice	Exercises, Portfolios

		G3.3	4		
	***B/* Homework and self-studying tasks**: *(6)* + Do lab exercises about design patterns			Group work, Case study, Lab practice	Exercises, Portfolios
15	***Course Review (2/2/6)***				
	A/ Main contents teaching in class *(4)* + Overall review + Exercises about software design through various stages	G2.1	3	Presentation, Group work, Case study	Exercises
	***B/* Homework and self-studying tasks**: *(6)* + Course review				

11. Assessments:

- Grading system: **10**
- Assessment plan:

Test type	Contents	Time	CLOs assessed	Comp. level	Method	Tools	Ratio (%)
Exercises							**30**
BT#1	Create usecase diagrams for different management systems.	Week 3	G1.1 G1.2 G2.1 G2.2	3 3 3 4	Case study, Lab exercises	Exercises, Portfolios	5
BT#2	Create sequence diagrams for different management systems.	Week 5	G1.1 G1.2 G2.1 G2.2	3 3 3 4	Case study, Lab exercises	Exercises, Portfolios	10
BT#3	Create class diagrams for different management systems.	Week 6	G1.1 G1.2 G2.1 G2.2	3 3 3 4	Case study, Lab exercises	Exercises, Portfolios	10
BT#4	Create activity diagrams for different management systems.	Week 8	G1.1 G1.2 G2.1 G2.2	3 3 3 4	Case study, Lab exercises	Exercises, Portfolios	5
Presentation							**20**

PP#1	Working in groups (3-4 students each) on one of the following topics, then give a presentation before class: 1. Factory method 2. Abstract Factory 3. Builder 4. Prototype 5. Singleton 6. Adapter 7. Bridge 8. Composite 9. Decorator 10. Command 11. Iterator 12. Observer 13. State 14. Strategy 15. Template method 16. Visitor	Week 9-15	G1.3, G2.1, G2.2, G3.1, G3.2, G3.3	3 3 4 5 5 4	Portfolios, Presentation	Rubrics, Portfolios	
Final project							**50**
BL#1	Working in groups (3-4 students each): Design a specific software system (assigned or chosen from a list) using various UML diagrams.	Week 5 – 15	G1.1, G1.2, G1.3, G2.1, G2.2, G3.1, G3.2, G3.3	3 3 3 3 4 5 5 4	Project-based, Portfolios	Rubrics, Portfolios	

CLOs	**Test types**					
	BT#1	BT#2	BT#3	BT#4	PP#1	BL#1
G1.1	X	X	X	X		X
G1.2	X	X	X	X		X
G1.3					X	X
G2.1	X	X	X	X	X	X
G2.2	X	X	X	X	X	X
G3.1					X	X
G3.2					X	X
G3.3					X	X

12. Learning materials:

Main textbooks:

[1] Kim Hamilton, Russell Miles, *Learning UML 2.0*, O'Reilly, 2006.

[2] E. Gamma, R. Helm, R. Johnson, J. Vlissides, *Design Patterns - Elements of Reusable Object-Oriented Software*, Addison-Wesley, 2005.

References:

[1] James W. Cooper, *The Design Patterns Java Companion*, Addison-Wesley, 1998.

[2] Horstmann, Cay, *Object-Oriented Design & Patterns*, Second Edition, John Wiley & Sons, 2006.

[3] Craig Larman, *Applying UML and Patterns: An Introduction to Object-oriented Analysis and Design and Iterative Development*, Pearson, 2008.

[4] http://www.omg.org/spec/UML/2.3/

13. Date of first approval:

14. Approval:

Dean	**Head of Department**	**Editor**

15. Updates:

1st update:	**Editor:** **Head of Department:**
2nd update:	**Editor:** **Head of Department:**