King Mongkut's University of Technology Thonburi
Department of Computer Engineering

Course Name: Object-oriented Analysis and Design
Number: CPE 372/CPE343
Semester: 2/2561 (January 2019)

Latest revision: 1 March 2019

Instructor Name: Dr. Sally Goldin
Office hours: Tuesdays 11:00-12:00, Thursdays 16:30-17:30 or by appointment
e-mail: se.goldin@goldin-rudahl.com

Course meeting times:
Lecture/Hands-on: Tuesdays 13:30-16:20 in Room 1115
Course website (lecture notes, assignments, etc.): http://windu.cpe.kmutt.ac.th/cpe372
Facebook group (announcements and discussion):
https://www.facebook.com/groups/CPE372.OOAD.2019

Catalogue Listing: Not currently in the CPE catalogue

Textbooks:

I have not been able to find a textbook I like that is available at a reasonable price in Thailand. I encourage students to use and/or purchase some of the references below.

For reference:

- M. Grand. *Patterns in Java, Volume 1*. New York: John Wiley & Sons, Inc., 1998
- M. Fowler. *UML Distilled: A brief guide to the standard object modeling language*. Addison-Wesley, 2003.
- [more references to be added later]

Course Description:
This course introduces students to the most important concepts underlying an object-oriented approach to software development, and provides practice in applying these concepts to create robust, maintainable, extensible programs. The course emphasizes object-oriented analysis and design skills, that is, the process of moving from functional requirements to a detailed specification of the classes comprising a software application and their interactions. Exercises and the term project will use the Java programming language, but this is not primarily a course about Java. Most of the course content is relevant to any object-oriented programming language.

Key Topics:
See detailed schedule below.

Prerequisites: CPE111/CPE113 – Students must have passed CPE111 or CPE113 with at least a grade of B in order to register for this class. Otherwise, they need the permission of the instructor.

Expected Prior Knowledge and Skills: English comprehension skills must be adequate to understand lecture material and assignment instructions and to communicate questions to the instructor. Students should be comfortable programming in C or some other procedural programming language, and familiar with basic programming concepts such as variables, functions, loops, conditionals, etc.

Learning Outcomes: Students who have completed this course should be able to:

1. Explain and use the concepts that distinguish an object-oriented programming approach from a procedural approach, such as encapsulation, inheritance, overloading, etc.
2. Given a set of functional requirements, apply OOD principles, techniques and tools (e.g. UML) to create and document the design for a software application.
3. Implement an object-oriented design to create a robust, extensible, maintainable application, using the Java programming language.
4. Given an object-oriented design created by someone else, critique the design and suggest improvements or alternative approaches.

Ethical Conduct: Students are expected to behave with honesty and integrity in completing assignments and doing exams. Consulting and sharing information with peers can be an effective way to learn, but copying simply to turn in an assignment or get a grade is unethical behavior. Similarly, allowing another student to copy your work is unethical. Behaving ethically will benefit the student in the long term by helping the student to become responsible for his or her own learning and growth, and by guaranteeing that others will view the student as honest and trustworthy.

Any student who cheats, on any assignment, lab, quiz, exam or project in this class, will be severely punished.

- First offense: the student will **lose 10 points off his or her final course grade**.
- Second offense: the student will **immediately fail the course**.

"Cheating" is defined as 1) copying someone else's answers; 2) submitting code or other work that was *completely or partially* created by someone else; 3) submitting false information about attendance; 4) allowing another student to completely or partially copy your work or answers. The decision of the instructor that a student has cheated is final.

Grading Policy:

- Exercises/homework: 15%
- Quizzes: 10%
- Class participation/critiques: 10%
- Programming project: 40% (20% design, 20% final application)
- Written final Exam: 25%

Quizzes will be given at the start of most lecture periods. Anyone who is late for the quiz will receive a zero score on that quiz.

Course Schedule

Holidays: Tuesday 19 February 2019 (Makha Bucha Day)
Monday 8 April 2019 (in lieu of Chakri Memorial Day)
Saturday 13 April to Friday 19 April 2019 (Songkran Holiday & break)
Monday 20 May 2019 (in lieu of Visakha Bucha Day)

Exam periods: Midterms Monday 4 to Tuesday 12 March
Finals Tuesday 14 to Friday 24 May

The details in the following table may change. Check the FB group for up-to-date information.

Week	Class Date	Topics and activities
1	17 January	What is OOD? Why use it?
2	24 January	Encapsulation and information hiding; classes and instances ***Students must form project teams***
3	31 January	*(Class was canceled)* ***Students must pick project topics***
4	7 February	Inheritance; abstraction and interfaces
5	14 February	Overloading, overriding and polymorphism
6	21 February	Introduction to UML: Use cases
7	28 February	Class diagrams
8	7 March	Sequence diagrams
9	14 March	Design patterns 1
10	21 March	Evaluating OO Designs
11	28 March	*No class – Instructor at a conference* ***Project design documents due by noon on 26 March***
12	4 April	Design presentations/critiques
13	11 April	Design presentations/critiques
14	18 April	*No class – Songkran holiday*
15	25 April	Design patterns 2
16	2 May	Refactoring and code reuse
17	9 May	Other object-oriented languages
18	16 May	Final examination - in class
18	23 May	***Projects due***