



Pontificia Universidad Católica de Chile
School of Engineering
Department of Computer Science

IIC2113 Detailed Software Design (II/2016)

Course Program

Professor	: Rodrigo Arturo Saffie Kattan
Email	: rasaffie@ing.puc.cl
Schedule	: Tuesdays and Thursdays M3 (11:30 - 12:50)
Assistantship schedule	: Fridays M6 (17:00 - 18:20)
Classroom	: Javier Pinto (DCC)
Prerequisites	: IIC2143
Assistants professors	: - Nebil Kawas (nakawas@uc.cl) - Patricio López (pelopez2@uc.cl) - Sebastián Salata (sasalata@uc.cl)

1 Objectives

At the end of the course, each student will be able to:

- Design and program object oriented software with expertise
- Assess the quality of a software design, through practical and theoretical criteria
- Analyze software in order to improve its efficiency, confiability and maintainability
- Use software design patterns when designing software

2 Contents

The following list contains some of the topics which will be studied throughout the course:

1. Introduction
 - (a) What is detailed software design?
 - (b) 4+1 architectural view model
2. Detailed software design principles
 - (a) Cohesion
 - (b) Coupling
 - (c) Abstraction

- (d) Encapsulation
- (e) SOLID
- 3. Detailed software design applied to components
 - (a) Software design patterns
 - (b) *Frameworks*
- 4. Testing and refactoring
 - (a) Testing techniques
 - (b) *Code Smells*
 - (c) Refactoring techniques
 - (d) Code quality metrics
- 5. Reverse engineering

3 Communication channels

The course will count with two official communication channels. Announcements and grades will be published through the course's official web page available at the SIDING platform. Everything else will be distributed through the course's Github organization.

4 Quizzes

The course considers periodical quizzes (C) about its contents and readings.

5 Presentations

The students of this course will have to do two oral presentations (P) about different subjects related with the course's contents. Such subjects will be announced with anticipation. Dates for presentations are:

First Presentation	October 13
Second Presentation	November 8

6 Assignments

The course considers three assignments (T) throughout the semester. Their publication and due dates are:

	Publication	Due
Assignment 1	August 25	September 9
Assignment 2	September 22	October 7
Assignment 3	October 27	November 11

7 Tests

The course considers two tests (I) throughout the semester. Their dates are:

Test 1	September 13
Test 2	October 18

8 Final Test

The course considers a final test (E), which is mandatory. It will be on November 25, at 8:30 am.

9 Grading

The course's final grade (N'_f) will be obtained with the following equation:

$$N'_f = 0.15 * \bar{C} + 0.15 * \bar{P} + 0.30 * \bar{T} + 0.20 * \bar{I} + 0.20 * E$$

In order to approve the course, the following conditions must be satisfied:

$$\bar{T} \geq 3.95$$

$$E \geq 2.95$$

$$N'_f \geq 3.95$$

Considering the above statements, the final grade will be:

$$N_f = \begin{cases} N'_f & \text{if all conditions are met} \\ \min\{3.9, N'_f\} & \text{if one or more conditions are not met} \end{cases}$$

10 Bibliography

- FOWLER, M., ET AL., *Patterns of Enterprise Application Architecture*, ADDISON-WESLEY, 2003.
- FOWLER, M., ET AL., *Refactoring: Improving the Design of Existing Code*, ADDISON-WESLEY, 2000.
- FOWLER, M., ET AL., *UML Distilled, (3^d ed.)*, ADDISON-WESLEY, 2004.
- GAMMA, E., ET AL., *Design Patterns: Elements of Reusable Object-Oriented Software, (1st ed.)*, ADDISON-WESLEY, 1994.
- MARTIN, R. C. *Clean Code: A Handbook of Agile Software Craftsmanship, (1st ed.)*, PRENTICE HALL, 2008.
- PRESSMAN, R. S. *Software Engineering: A Practitioner's Approach, (7th ed.)*, MCGRAW-HILL EDUCATION, 2009.

11 Academic Integrity Policy

The course adheres to the University's official academic integrity policy, which can be found in the following link.