CPIT-252 Syllabus

Catalog Description

CPIT-252 Software Design Patterns
Credit: 3 (Theory: 3, Lab: 0, Practical: 1)
Prerequisite: CPIT-251
Classification: Department Required

The objective of this course is to study the principles behind the patterns of software and to then apply a number of basic patterns. This course covers fundamental aspects of large scale software architecture, defined frameworks, design patterns, and ways of developing and establishing systems based on components. The purpose of this course is: (1) to know the classical styles of software pattern and the need for a language to describe the architecture, (2) to understand how to express the qualities we want our architecture to provide to the system or systems we are building from it, and (3) to know how to achieve software qualities using TACTICS. Topics include envisioning architecture (architecture business cycle), architectural patterns, reference models, reference architectures, understanding quality attributes, achieving qualities using tactics, and how to document software architecture.

Class Schedule

Lab/Tutorial 90 minutes 1 times/week

Meet 50 minutes 3 times/week or 80 minutes 2 times/week

Textbook

Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, , "Design Patterns", Addison-Wesley; 2 edition (2003-09)

ISBN-13 9780582844421 **ISBN-10** 0582844428

Grade Distribution

Week	Assessment	Grade %
3	Graded Lab Work 1	2
5	Exam 1	15
5	Graded Lab Work 2	3
7	Graded Lab Work 3	4
10	Exam 2	15
10	Graded Lab Work 4	4
12	Graded Lab Work 5	2
15	Lab Exam	5
15	Group Project	20
16	Comprehensive Final Exam	30

Last Articulated

December 18, 2017

Relationship to Student Outcomes

a	b	c	d	e	f	g	h	i	j	k	l	m	n
	x	x						x		x	x	x	

Course Learning Outcomes (CLO)

By completion of the course the students should be able to

1. Generate code from class diagrams and vice versa (i)
2. Identify any GoF creational design patterns for code or scenario or design (k)
3. Design and implement GoF creational design patterns (c)
4. Identify any GoF structural design patterns for code or scenario or design (k)
5. Design and implement GoF structural design patterns (c)
6. Identify any GoF behavioural design patterns for code or scenario or design (k)
7. Design and implement GoF behavioural design patterns (c)
8. **Desing and Implement an application that simulate a medium size project as part of a group (l)**
9. **Detect the appropriate design patterns in class diagrams (b)**
10. **Recognize the design patterns from code (k)**
11. **Produce a pattern based design that maps the requirements of a given scenario using design patterns (k)**
12. **Implement code that maps the requirements of a given scenario or designs (c)**
13. Evaluate the quality of software design using ISO25001 standard (m)

Coordinator(s)

Dr. Georgios Tsaramirsis, Associate Professor

CPIT-252 Syllabus

Topics Coverage Durations

Topics	Weeks
Introduction	1
UML2 Class diagrams	2
Overview to Design Pattens	1
Creational Patterns	2
Structural Patterns	3
Behavioral Patterns	4
Software Quality	2