University of Macau
Computer and Information Science Department
SFTW372 – Object Oriented Analysis and Design Patterns
Syllabus
2nd Semester 2011/2012
Part A – Course Outline

Elective course in Computer Science

Catalog description:

(3-2) 4 credits. The course discusses object-oriented analysis and design using Unified Modeling Language (UML). The main contents are use case diagram, class diagram, sequence diagram, state diagram, activity diagram, component diagram and deployment diagram of UML. And Object Constraint Language (OCL) and design patterns are also discussed. CASE tool of UML is used to analyze and design the course project systems.

Course type:

Theoretical with substantial laboratory/practice content

Prerequisites:

- None

Textbook:

- Craig Larman: ***Applying UML and Patterns***, 3rd ed. Prentice-hall, 2005.

References:

- J. Arlow and I. Neustadt: ***UML2 and the Unified Process: Practical Object-Oriented Analysis and Design***, 2nd ed. Addison Wesley, 2005.
- G. Booch, J. Rumbaugh, and I. Jacobson: ***The Unified Modeling Language User Guide*****,** Addison-Wesley, 1998**.**
- E. Gamma, R. Helm, R. Johnson and J. Vlissides: ***Design Patterns: Elements of Reusable Object-Oriented Software,*** Addison-Wesley, 1995.

Major prerequisites by topic:

1. Basic concepts of object-oriented programming.

Course objectives:

1. Analyze and design with object-oriented method in UML[a, b, c, e, g, k]
2. Describe constraints and introduce OCL. [a, b, c, e, g, k]
3. Introduce design pattern technology. [a, b, c, e, g, k]
4. Apply object-oriented technology to the practical system analysis and design [a, b, c, e, g, k]

Topics covered:

1. **Introduction of OOA/D with UML (6 hours):** Introduce the concepts of object-orientation, object-oriented and analysis, Unified Modeling Language (UML), and applying OOA/D with UML into a practical case study.
2. **Unified Development Processes (3 hours):** Introduce the concepts of software development process, process activities, and Unified Development Process.
3. Requirements Modeling (9 hours): Analyze and specify the requirements model, including use case diagram, use case definition, system operation sequence diagram, activity diagram, operation contract with pre and post conditions, and conceptual class diagram by illustrating with case studies.
4. **Static Design Modeling (6 hours):** Design system static model, including design class diagram, identification of class, attribute and methods, identifying generalization, aggregation, composition, and dependency relations, specifying multiplicities and constraints of associations by illustrating with case studies.

5. **Dynamic Design Modeling (9 hours):** Design system dynamic model, including design sequence diagram, activity diagram and state diagram, mapping design to codes, component and deployment diagrams, by illustrating with case studies.
6. **OCL (3 hours)**: Introduce Object Constraint Language, including specifying pre, post conditions and invariants with OCL by illustrating with case studies.
7. **Design Patterns (3 hours):** Introduce the GRASP and GoF patterns, including the concept of design patterns, designing objects with responsibility, expert pattern, create pattern, controller, the concepts of low coupling and high cohesion, façade and adapter patterns.

Class/laboratory schedule:

Timetabled work in hours per week			No of teaching weeks	Total hours	Total credits	No/Duration of exam papers
Lecture	Tutorial	Practice				
3	2	Nil	14	70	4	1 / 3 hours

Student study effort required:

Class contact:	
Lecture	42 hours
Tutorial	28 hours
Other study effort	
Self-study	36 hours
Homework assignment	10 hours
Project / Case study	20 hours
Total student study effort	136 hours

Student assessment:
Final assessment will be determined on the basis of:

Homework	20%	Project	30%
Mid-term	20%	Final exam	30%

Course assessment:
The assessment of course objectives will be determined on the basis of:

- Homework, project and exams
- Course evaluation

Course outline:

Weeks	Topic	Course work
1,2	**Introduction of OOA/D with UML** The concepts of object-orientation, object-oriented and analysis, Unified Modeling Language (UML), and applying OOA/D with UML into a practical case study.	
3	**Unified Development Processes** Software development process, process activities, and Unified Development Process.	Assignment#1 Course Project Introduction
4,5,6	**Requirements Modeling** Requirements model, use case diagram, use case definition, system operation sequence diagram, activity diagram, operation contract with pre and post conditions, and conceptual class diagram, and case studies.	Assignment#2 Requirements Model

7,8	**Static Design Modeling** Design class diagram, identification of class, attribute and methods, generalization, aggregation, composition, and dependency relations, association, multiplicity and constraint.	Assignment#3 Static Design Model
9,10,11	**Dynamic Design Modeling** Design sequence diagram, activity diagram and state diagram, mapping design to codes, component and deployment diagrams, and case studies.	Assignment#4 Dynamic Design Model
12	**OCL** Object Constraint Language, formal specifying pre, post conditions and invariants with OCL.	
13	**Design Patterns** Design patterns, designing objects with responsibility, expert pattern, create pattern, controller, low coupling and high cohesion, façade and adapter patterns.	Assignment#5 Apply OCL and Patterns to Project
14	**Project Presentation**	Course Project Report

Contribution of course to meet the professional component:

This course prepares students to work professionally in the area of software development.

Relationship to CS program objectives and outcomes:

This course primarily contributes to Computer Science program outcomes that develop student abilities to:
(a) an ability to apply knowledge of computing, science, and engineering.
(b) an ability to design and conduct experiments, as well as to analyze and interpret data.
(c) an ability to design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability.
(e) an ability to identify, formulate, and solve engineering problems.
(g) an ability to communicate effectively
(k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.

Relationship to CS program criteria:

Criterion	DS	PF	AL	AR	OS	NC	PL	HC	GV	IS	IM	SP	SE	CN
Scale: 1 (highest) to 4 (lowest)		2					2				3		1	

Discrete Structures (DS), Programming Fundamentals (PF), Algorithms and Complexity (AL), Architecture and Organization (AR), Operating Systems (OS), Net-Centric Computing (NC), Programming Languages (PL), Human-Computer Interaction (HC), Graphics and Visual Computing (GV), Intelligent Systems (IS), Information Management (IM), Social and Professional Issues (SP), Software Engineering (SE), Computational Science (CN).

Course content distribution:

Percentage content for			
Mathematics	Science and engineering subjects	Complementary electives	**Total**
0%	100%	0%	100%

Coordinator:

Prof. Zhi Guo Gong

Persons who prepared this description:

Prof. Xiao Shan Li, Dr. Fai Wong

Part B General Course Information and Policies

2nd Semester 2011/2012

Instructor:	Prof. Xiaoshan Li	Office:	N421
Office Hour:	by appointment	Phone:	8397-4471
Email:	xsl@umac.mo		

Time/Venue: (*to be announced*)

Grading Distribution:

Percentage Grade	Final Grade	Percentage Grade	Final Grade
100 – 93	A	92 - 88	A–
87 – 83	B+	82 - 78	B
77 – 73	B–	72 - 68	C+
67 – 63	C	62 - 58	C–
57 – 53	D+	52 - 50	D
below 50	F		

Comment:
The objectives of the lectures are to explain and to supplement the text material. Students are responsible for the assigned material whether or not it is covered in the lecture. Students who wish to succeed in this course should read the assignments prior to the lecture and should work all homework and project assignments. You are encouraged to look at other sources (other texts, etc.) to complement the lectures and text.

Homework and Course Project:
The project is the important part of this course. Through the project, students can apply the course contents they learn to the practical software system analysis and design. It will be very helpful for them to improve the analysis and design ability of object-oriented method. Project progress stage reports are requested to be delivered as homework assignments round two or three weeks with the progress of course contents course, and are presented, discussed and commented during tutorial classes and in instructor's office outside of class. Finally, each team should deliver their final course project report at the end of semester before final exam.

- The requirements will be announced and discussed in class.
- The students' progress on their project will be discussed in the tutorial class and instructor's office.
- The project will be presented twice formally at the middle and end of semester, and the final project report should be delivered before the final exam.

Note

- Recitation session is important part of this course and attendance is strongly recommended.
- Check UMMoodle (https://ummoodle.umac.mo/) for announcement, homework and lectures. Report any mistake on your grades within one week after posting.
- No make-up exam is given except for CLEAR medical proof.
- Cheating is absolutely prohibited by the university.

Rubric for Program Outcomes

Rubric for (a)	5 (Excellent)	3 (Average)	1 (Poor)
Use a correct model and formulation correctly	Students choose a model correctly and properly apply correct techniques.	Students choose a wrong model sometime, use a wrong formula, or a different technique.	Students use a wrong model and wrong formula, or do not know how to model.
Rubric for (b)	**5 (Excellent)**	**3 (Average)**	**1 (Poor)**
Design experiments	Student understands what needs to be tested and designs an appropriate experiment that takes into account the limitations of the equipment and measurement accuracy.	Student understands what needs to be tested and designs an appropriate experiment, but may not fully understand the limitations of the measurements.	Student does not understand what needs to be tested and/or does not design an appropriate experiment.
Rubric for (c)	**5 (Excellent)**	**3 (Average)**	**1 (Poor)**
Design capability and design constraints	Student understands very clearly what needs to be designed and the realistic design constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability.	Student understands what needs to be designed and the design constraints, but may not fully understand the limitations of the design constraints.	Student does not understand what needs to be designed and the design constraints.
Rubric for (e)	**5 (Excellent)**	**3 (Average)**	**1 (Poor)**
Modeling, problem formulation and problem solving	Students choose and properly apply the correct techniques.	Students model correctly but cannot select proper technique or model incorrectly but solve correctly accordingly.	Students at loss as to how to solve a problem.
Rubric for (g)	**5 (Excellent)**	**3 (Average)**	**1 (Poor)**
Written component	Document is nearly error free with sophisticated use of vocabulary, formatted properly, with well-developed concise sentences and paragraphs.	Document contains some errors with a somewhat colloquial vocabulary, minor formatting issues, with some organizational issues that do not interfere with communication.	Document contains many errors, very colloquial vocabulary, with severe organizational issues that interfere with communication. Document would be considered unacceptable.
Rubric for (k)	**5 (Excellent)**	**3 (Average)**	**1 (Poor)**
Use modern principles, skills, and tools in engineering practice	Student applies the principles, skills and tools to correctly model and analyze engineering problems, and understands the limitations.	Student applies the principles, skills and tools to analyze and implement engineering problems.	Student does not apply principles and tools correctly and/or does not correctly interpret the results.