University of Macau
Computer and Information Science Department
SFTW440 – Software Engineering Principles
Syllabus
1st Semester 2012/2013
Part A – Course Outline

Elective course in Computer Science

Catalog description:
(3-2) 4 credits. The course discusses the theories, methods and tools of software engineering for developing large and complex software systems. The main contents are requirement specification, system modeling, architectural design, object-oriented analysis and design, verification and validation, and software testing. The Unified Modeling Language (UML) and its CASE tool are used to analyze and design the course project systems.

Course type:
Theoretical with substantial laboratory/practice content

Prerequisites:
None

Textbook(s) and other required material:
- Ian Sommerville: ***Software Engineering***, 9th ed. Addison-Wesley, 2011

References:
- Roger S. Pressman: ***Software Engineering: A Practitioner's Approach,*** 7th ed. McGraw-Hill, 2009.
- L. Maciaszek and B. Liong: ***Practical Software Engineering: A Case Study Approach***, Pearson Education, 2005.
- Craig Larman: ***Applying UML and Patterns***, 3rd ed. Prentice-hall, 2005.
- J. Arlow and I. Neustadt: ***UML2 and the Unified Process: Practical Object-Oriented Analysis and Design***, 2nd ed. Addison Wesley, 2005

Major prerequisites by topic:
None

Course objectives*:
1. Introduce the concepts of software engineering. [a, b, c, e]
2. Specify and analyze requirements of software system. [a, b, c, e, g]
3. Model software system design with UML. [a, b, c, e, g, k]
4. Verify and validate software system. [a, b, c, e, g]

Topics covered:
- **Introduction of Software Engineering (6 hours)**: Introduce the concepts of software, software engineering, software process, software process model, software costs, attributes of good software, software engineering methods and key challenges, professional and ethical responsibility, emergent system properties, system engineering, legacy systems, and system dependability.
- **Software development processes (3 hours)**: Discuss software development process models, process iteration, process activities, Rational Unified Process, and Computer-Added Software Engineering (CASE).

- **Software Requirements Analysis and Specification (6 hours)**: Analyze and specify the requirements model, including functional and non-functional requirements, requirements engineering processes, requirements elicitation and analysis, requirements definition and specification, requirements validation and management, system models, use case diagram and conceptual class diagram, use case definition, pre and post condition and constraints.
- **Architectural Design (6 hours)**: Study the architectural design of software systems, including system architectural design decisions and views, architectural patterns and application architectures.
- **System Analysis, Design and Implementation (9 hours)**: Apply object-oriented method to developing software systems, including object concept and class identification, object-oriented design using UML, design patterns and implementation issues.
- **Verification and Validation (3 hours)**: Understand the concepts of software verification and validation, software inspection, automated static analysis, verification and formal methods, clean-room software development.
- **Software Testing (3 hours)**: Study the software testing, including system testing, component testing, white and black box testing, test case design and test automation.

Note: the topics of software project management are covered by a compulsive course SFTW497: Software Project Management.

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 Software Engineering** Concepts of software, software engineering, software process and model, software costs and attributes, software engineering methods and key challenges, emergent system properties, and system dependability.	Review UML and use its CASE tool
3	**Software Development Processes** Software development process models, process iteration, process activities, Rational unified process, and Computer-Added Software Engineering (CASE).	Assignment#1 & Course Project Instruction
4-5	**Software Requirements Analysis and Specification** Requirements model, functional and non-functional requirements, requirements engineering process, requirements elicitation and analysis, requirements definition, specification, validation and management.	Assignment#2 & Project Requirements Model
6-7	**Architectural Design** Architectural design decisions and views, architectural patterns and application architectures	Assignment#3 Project Requirements Model (revised)
8	**Mid-Project Presentation** Presentation of the course project requirements models	Midterm Exam
9 -11	**System Analysis, Design and Implementation** Object concept and class identification, object-oriented design using UML, design patterns and implementation issues.	Assignment#4 & Project Design Model
12-13	**Verification and Validation, and Software Testing** The concepts of software verification and validation, software inspection, automated static analysis, verification and formal methods, clean-room software development, System and component testing, white and black box testing, test case design and test automation.	Assignment#5 & Project System Validation and Testing
14	**Final-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

1st Semester 2011/2012

Instructor:	Prof Xiaoshan Li	Office:	N421
Office Hour:	Mon., Wed., Thur., & Fri.:14:00 – 17:30pm	Phone:	8397-4471
Email:	xsl@umac.mo		

Time/Venue: Tue. 16:30 – 17:30pm, ILG206 (lectures)
Sat. 08:30 – 10:30am, J406 (practice/tutorials)

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 the part of 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.