University of Macau
Faculty of Science and Technology
Department of Computer and Information Science
SFTW422 Logic and Computability
Syllabus
2nd Semester 2011/2012
Part A – Course Outline

Elective course in Computer Science

Course description:
(3-2) 4 credits. Introduction to logic, propositional calculus and predicate calculus, resolution and logic programming, temporal logic and formalization of programs, also computability will be introduced.

Course type:
Theoretical

Prerequisites:
- SFTW223

Textbook(s) and other required material:
- M. Ben-Ari. (2001) *Mathematical Logic for Computer Science,* 2nd ed., Springer.

References:
- E. Mevdelson. (1987) *Introduction to Mathematical Logic.* 3rd ed., Wadsworth & Brooks/Cle.
- C. C. Leary. (2000) *A Friendly Introduction to Mathematical Logic.* Prentice Hall.

Major prerequisites by topic:
- Basic concepts of propositional logic.
- Discrete mathematics.
- Elements of theory of computation.
- Methods for constructing mathematical proofs.

Course objectives:
- Introduce formal system (propositional and predicate logic) which mathematical reasoning is based on (a)
- Develop an understanding of how to read and construct valid mathematical arguments (proofs) (a, e)
- Understand mathematical statements (theorems), including inductive proofs (a, e)
- Introduce and work with various problem solving strategies and techniques (a, e, k)

Topics covered:
- **Basic Concepts (2 hours)**: Logic and Reasoning.
- **Formal Systems (2 hours)**: Propositional Logic and Predicate Logic.
- **Propositional Calculus (10 hours)**: Syntax, Semantics, Tableaux, Equivalence, Satisfiability, Validity and Consequence, Deductive Proofs and Resolution.
- **Predicate Calculus (12 hours)**: Syntax, Semantics, Tableaux, Interpretations, Satisfiability, Deduction and Resolution.
- **Propositional Temporal Logic (8 hours)**: Syntax, Semantics, Tableaux, Soundness and Completeness.
- **Formalization of Programs (6 hours)**: Axiomatization of Language, Formal Specification, Formal Methods and Formal Verification of Programs.

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	25 hours
Homework assignment	15 hours
Total student study effort	110 hours

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

Attendance	5%	Homework	10%		
1st Mid-term	25%	2nd Mid-term	25%	Final exam	35%

Course assessment:
The assessment of course objectives will be determined on the basis of:
- Homework and exams
- Course evaluation

Course outline:

Weeks	Topic	Course work
1	**Introduction** Mathematical basis of software, morphology, propositional calculus, generalization of propositional calculus, logic programming, non-standard logic	
2-4	**Propositional Calculus** Operators, formulas, interpretations, satisfiability, validity, semantic tableaux, deductive proofs, Gentzen systems, Hilbert systems, resolution	Assignment 1,2
5-7	**Predicate Calculus** Relations, predicates, interpretations, logic equivalence, substitution, semantic tableaux, Gentzen system Hilbert system, functions, clausal form, Herbrand models	Midterm 1 Assignment 3
8-10	**Ground Resolution** Substitution, unification, most general unifier, unification algorithms, general resolution, theorem proving, logic programming	Assignment 4
11-12	**Temporal Logic** Syntax and semantics of PTL, deductive system for linear PTL, sematic tableaux, soundness and completeness	Midterm 2 Assignment 5
13-14	**Formalization of Programs** Axiomalization of language, deductive system for Hoare Logic, proving programs	Assignment 6

Contribution of course to meet the professional component:
This course prepares students with fundamental knowledge of mathematical logic for computer science.

Relationship to CS program objectives and outcomes:
This course primarily contributes to the Computer Science program outcomes that develop student abilities to:
(a) an ability to apply knowledge of mathematics, science, and engineering.
(e) an ability to identify, formulate, and solve engineering problems.

This course secondarily contributes to the Computer Science program outcomes that develop student abilities to:
(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)	1		2										4	

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**
50%	50%	0%	100%

Coordinator:
Dr. Fai Wong

Persons who prepared this description:
Dr. Fai Wong, Prof. Yi Ping Li

Part B – General Course Information and Policies

2nd Semester 2011/2012

Instructor: Prof. Yi Ping Li
Office: N411
Office hour: *to be announced*
Phone: 8397 4461
Email: ypli@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 textbook prior to the lecture and should work all homework assignments. You are encouraged to look at other sources (other texts, etc.) to complement the lectures and text.

Homework policy:
The completion and correction of homework is a powerful learning experience; therefore:
- There will be approximately 6 homework assignments.
- Homework is due one week after assignment unless otherwise noted, no late homework is accepted.
- Homework must be neatly typed and printed on a word-processor.
- The course grade will be based on the average of the HW grades.

Exams:
Two mid-term exams will be held during the semester.

Note:
- 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.

Appendix:

Rubric for Program Outcomes

Rubric for (a)	5 (Excellent)	3 (Average)	1 (Poor)
Understand the theoretic background	Students understand theoretic background and the limitations of the respective applications.	Students have some confusion on some background or do not understand theoretic background completely.	Students do not understand the background or do not study at all.
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 (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.