University of Macau
Faculty of Science and Technology
Department of Computer and Information Science
SFTW342 Programming Languages Architecture II
Syllabus
1st Semester 2012/2013
Part A – Course Outline

Elective course in Computer Science

Course description:
(3-2) 4 credits. This is the second of a 2-course sequence (SFTW241 compulsory + SFTW342 optional) developing the concepts, techniques, and models of computer programming established in SFTW241. The course revisits object-oriented models and develops student understanding in fundamental syntactic and semantic issues of language design and specification. Example languages used to illustrate language design include Java and script languages. In addition, the course design includes multimedia programming and modern programming environments for web programming and mobile programming.

Course type:
Theoretical with substantial laboratory/practice content

Prerequisites:
- SFTW241

Textbook(s) and other required material:
- Mark Guzdial and Barbara Ericson. (2007). *Introduction to Computing & Programming with Java: A Multimedia Approach*. Pearson Prentice Hall.

References:
- Bruce Eckel. (2006). *Thinking in Java (Fourth Edition).* Pearson Education, Inc.
- Gail Rahn Frederick and Rajesh Lal. (2009). *Beginning Smartphone Web Development: Building JavaScript, CSS, HTML, and Ajax-Based Applications for iPhone, Android, Palm Pre, BlackBerry, Windows Mobile, and Nokia S60.* APress.

Major prerequisites by topic:
- Language proficiency in at least one procedural language, ANSI C, and one object-oriented language, C++
- Modular and object-oriented programming techniques, with abstract data types and separate compilation method.
- Fundamental algorithmic analysis and design of different programming constructs and data structures.

Course objectives:
- Introduce to students the general principles and techniques of multimedia programming, including how to manipulate the pictures, digital sounds, and movies etc. [a, e ,k]
- Learn core issues of Java programming languages, including control flow, data types, methods and classes, etc. [a, e]
- Familiar with modern programming environments for web programming and mobile programming. [a, k]

Topics covered:
- **Introduction of general principles and techniques in picture programming (15 hours):** Introduce the basic knowledge of picture representations in computer, the general principles and

programming techniques in changing picture color values, copying and transforming pictures, edge detection, drawing using graphics class, etc.

- **Introduction of general principles and techniques in sound programming (9 hours):** Introduce the basic knowledge of sound coding in computer, the general principles and programming techniques in physics of sound, changing the volume of sounds, normalizing sounds, blending sounds, and music synthesis, etc.
- **Introduction of creating and modifying text for web (9 hours):** Introduce the general principles and programming techniques including writing programs to general HTML, using a database to build web pages, and handling the text for the web, etc.
- **Introduction of general principles and techniques in movie programming (3 hours):** Introduce the basic knowledge and programming techniques in video editing, including encoding, manipulating, and creating movies.
- **General introduction of mobile web development (6 hours):** Introduce the basic knowledge of the mobile industry, mobile web environment and mobile web development.

Class/laboratory schedule:

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

Student study effort required:

Class contact:	
Lecture	42 hours
Tutorial	14 hours
Practice	14 hours
Other study effort	
Self-study	14 hours
Assignment and projects	28 hours
Total student study effort	112 hours

Student assessment:

Final assessment will be determined on the basis of:

Assignment	20%	Lab exercises and reports	10%
Project	20%	Mid-term Exam	20%
Final exam	30%		

Course assessment:

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

- Assignments, lab exercises, project and examinations.
- Course evaluation

Course outline:

Weeks	Topic	Course work
1-5	**Introduction of general principles and techniques in picture programming** Java based picture manipulation, including modifying pictures using loop, modifying pixels in a matrix, edge detection and drawing, etc.	1. Assignment 1 Part I & II 2. Lab exercises 1-3
6-8	**Introduction of general principles and techniques in sound programming** Java based sound manipulation, including sound encoded, changing the volume of sounds, normalizing sounds, manipulating	1. Assignment 2 Part I 2. Lab exercises 4-6

Weeks	Topic	Course work
	different sections of sound differently, blending sounds, creating an echo, and additive synthesis, etc.	
9-11	**Introduction of creating and modifying text for web** Java based text modification for web, including creating and modifying text, making text for the web, writing programs to general HTML, using a database to build web pages, and handling the text for the web, etc.	1. Assignment 2 Part II 2. Lab exercises 7,8
12	**Introduction of general principles and techniques in movie programming** Java based movie manipulation, including generating frame-based animations, working with video frames, encoding and manipulating movies, etc.	1. Project 2. Lab exercise 9
13-14	**General introduction of mobile web development** Including mobile web vs. desktop web, mobile markup languages introduction, mobile scripting languages introduction, mobile industry groups and standards bodies, the mobile ecosystem, and mobile development tools introduction, etc.	Lab exercise 10

Contribution of course to meet the professional component:
This course prepares students to work professionally in the area of multimedia programming.

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 computing, mathematics, science, and engineering.
(e) an ability to analyze a problem, and identify, formulate and use the appropriate application requirements for obtaining its computing solution.

The course secondarily contributes to the Computer Science program outcomes that develop student abilities to:
(k) an ability to use the techniques, skills, and modern computer 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)		3	4	1			2							

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%

Persons who prepared this description:
Dr. Li Ming Zhang

Part B – General Course Information and Policies

1st Semester 2012/2013

Instructor: Dr. Liming Zhang Office: B1-B703
Office hour: Wed 2:30 pm – 5:30 pm, or by appointment Phone: 8397 8467
Email: lmzhang@umac.mo

Time/Venue: Tue 9:30 pm – 12:30 pm, (lectures)
Thurs 2:30 am – 4:30 pm, (laboratory)

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 work all assignments, lab exercises and project. Students 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 2 homework assignments, 10 lab exercises, and 1 project.
- Assignments and project are due four weeks, the lab exercises are due one week after assignment unless otherwise noted, late homework will be deducted 10% marks for each delayed day.
- Possible revision of homework grades may be discussed with the grader within one week from the return of the marked homework.
- The course grade will be based on the average of the homework grades.

Exams:
One 2-hour mid-term exam will be held during the semester. One 3-hour final exam will be held at the end of semester. Both the mid-term and final exams are closed book examinations.

Note:
- Lab exercise sessions are important part of this course and attendance is strongly recommended.
- Check UMMoodle 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)**
Identify applications in engineering systems	Students understand problem and can identify fundamental formulation.	Students understand problem but cannot apply formulation, or cannot understand problem.	Students cannot identify correct terms for engineering applications.
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.