City University of Hong Kong
Course Syllabus

offered by Department of Computer Science
with effect from Semester A 2018/19

Part I Course Overview

Course Title:	Advanced Programming
Course Code:	CS3391
Course Duration:	One semester
Credit Units:	3 credits
Level:	B3
Proposed Area: *(for GE courses only)*	☐ Arts and Humanities ☐ Study of Societies, Social and Business Organisations ☐ Science and Technology
Medium of Instruction:	English
Medium of Assessment:	English
Prerequisites: *(Course Code and Title)*	CS2310 Computer Programming or CS2311 Computer Programming or CS2313 Computer Programming or equivalent
Precursors: *(Course Code and Title)*	Nil
Equivalent Courses: *(Course Code and Title)*	Nil
Exclusive Courses: *(Course Code and Title)*	Nil

Part II Course Details

1. Abstract

(A 150-word description about the course)

For many algorithmic problems found in practical systems, the best solutions are usually the elegant combinations of both efficient algorithms and advanced programming techniques. They are the results of some exciting blend of programming, mathematics and problem solving. The course introduces an interesting variety of subjects in programming, algorithms, and discrete mathematics through puzzles and practical problems so that students will have the chance to perform original discovery of new programming challenges and devise new ideas on solving the problems in an innovative way in terms of algorithms and programming techniques. The focus of this course is to help students develop advanced algorithmic and programming skills that are required to solve sophisticated problems in the real world. Due to the practicality of the problems which appear in many collegiate programming contests, we expect that the best students from this course will also be competent to solve competition-style programming problems and may be able to represent City University of Hong Kong at ACM Collegiate Programming contests.

2. Course Intended Learning Outcomes (CILOs)

(CILOs state what the student is expected to be able to do at the end of the course according to a given standard of performance.)

No.	CILOs[#]	Weighting* (if applicable)	Discovery-enriched curriculum related learning outcomes (please tick where appropriate)		
			A1	*A2*	*A3*
1.	Analyze programming problems, create new ideas on constructing algorithms and propose programming techniques for solving the problems.	40%			
2.	Write computer programs based on the algorithms devised and programming techniques chosen for solving problems.	40%	✓	✓	
3.	Write computer programs to solve problems under time pressure.	5%		✓	
4.	Generate new approaches on enhancing team programming and problem solving techniques.	15%			
	** If weighting is assigned to CILOs, they should add up to 100%.*	100%			

[#] Please specify the alignment of CILOs to the Gateway Education Programme Intended Learning outcomes (PILOs) in Section A of Annex.

A1: Attitude
Develop an attitude of discovery/innovation/creativity, as demonstrated by students possessing a strong sense of curiosity, asking questions actively, challenging assumptions or engaging in inquiry together with teachers.

A2: Ability
Develop the ability/skill needed to discover/innovate/create, as demonstrated by students possessing critical thinking skills to assess ideas, acquiring research skills, synthesizing knowledge across disciplines or applying academic knowledge to self-life problems.

A3: *Accomplishments*
Demonstrate accomplishment of discovery/innovation/creativity through producing /constructing creative works/new artefacts, effective solutions to real-life problems or new processes.

3. Teaching and Learning Activities (TLAs)

(TLAs designed to facilitate students' achievement of the CILOs.)

Teaching pattern:

Suggested lecture/laboratory mix: 3 hrs. workshop

TLA	Brief Description	CILO No.				Hours/week (if applicable)
		1	2	3	4	
Lecture	Algorithms and related examples are introduced and studied in lectures.	✓				
Lab	Students can create new ideas and invent new approaches on designing algorithms and computer codes to solve programming problems during the laboratory sessions. In the laboratory sessions, sets of problems are given to students. The students can learn how to analyze the problems and devise strategies with optimal use of resources and time to solve the problems. Based on the algorithms and programming techniques proposed by students or given by instructors, students spend the laboratory sessions to instantiate the algorithms by writing corresponding computer programs. In regular laboratory sessions, students tackle programming problems in teams. Various team formations will be arranged so that students can work with different individuals, so to generate new approaches on enhancing team programming and problem solving techniques.		✓	✓	✓	
Assignments	Take-home assignments also help students improve their proficiency on various programming techniques.		✓			
Quiz	Students will compete with each other and try to finish as many problems as possible within a limited period of time.			✓		

4. Assessment Tasks/Activities (ATs)

(ATs are designed to assess how well the students achieve the CILOs.)

Assessment Tasks/Activities	CILO No.				Weighting*	Remarks
	1	2	3	4		
Continuous Assessment: 60%						
Assignments	✓	✓			45%	
Quiz			✓	✓	15%	May work individually or in teams
Examination			✓		40%	
Examination^: 40% (duration: 3 hours)						
** The weightings should add up to 100%.*					100%	

^ For a student to pass the course, at least 30% of the maximum mark for the examination must be obtained.

5. Assessment Rubrics

(Grading of student achievements is based on student performance in assessment tasks/activities with the following rubrics.)

Assessment Task	Criterion	Excellent (A+, A, A-)	Good (B+, B, B-)	Fair (C+, C, C-)	Marginal (D)	Failure (F)
1. Assignments	ABILITY to SOLVE questions using the techniques learned in the lectures	High	Significant	Moderate	Basic	Not even reaching marginal levels
2. Quiz	ABILITY to solve problems under time pressure and group collaboration	High	Significant	Moderate	Basic	Not even reaching marginal levels
3. Examination	ABILITY to solve problems for different topics under time pressure	High	Significant	Moderate	Basic	Not even reaching marginal levels

Part III Other Information (more details can be provided separately in the teaching plan)

1. Keyword Syllabus

(An indication of the key topics of the course.)

Standard libraries in C and C++; C and C++ input and output processing; Recursion; Dynamic programming; Parsing; Graph algorithms; Strings; Search algorithms; Simulation problems; Spanning trees; Sets; Shortest path; Maximum flow; Computational geometry; Arithmetic, Algebra and number theory; Greedy and enumeration algorithms.

2. Reading List

2.1 Compulsory Readings

(Compulsory readings can include books, book chapters, or journal/magazine articles. There are also collections of e-books, e-journals available from the CityU Library.)

1.	Steven S. Skiena and Miguel A. Revilla (2003). *Programming Challenges: The Programming Contest Training Manual*, Springer-Verlag.

2.2 Additional Readings

(Additional references for students to learn to expand their knowledge about the subject.)

1.	Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms.* McGraw Hill, first or second edition.
2.	Steven Halim (2013). *Competitive Programming 3.* Lulu.