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:	Data Structures
Course Code:	CS3334
Course Duration:	1 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 CS2312 Problem Solving and 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)

This course aims to provide students an appreciation to the fundamentals of computer science. Models and applications of data structures including heaps, search trees, hash tables and disjoint sets are introduced and evaluated. Mathematical tools for analysis of algorithms and data structures are discussed and applied. Students are given the opportunity to develop and implement applications of the data structures and their derivatives.

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.	Implement common data structures and algorithms.			✓	
2.	Analyse efficiency and correctness of algorithms using mathematical techniques.			✓	
3.	Evaluate and compare similar data structures and algorithms.		✓	✓	
4.	Design and apply appropriate data structures to solve problems.		✓	✓	
** 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/tutorial/laboratory mix: 2 hrs. lecture; 1 hr. tutorial.

TLA	Brief Description	CILO No.				Hours/week (if applicable)
		1	2	3	4	
Lecture	Explain key concepts about algorithms and data structures for searching, indexing, sorting, manipulating data.	✓	✓	✓	✓	
Tutorial	Work on hands-on exercises and labs related to the key concepts taught in lectures.	✓	✓	✓	✓	
Assignments	Require students to do programming and analysis tasks.	✓	✓	✓	✓	

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^: 40%						
Assignments	✓	✓	✓	✓	25%	Students are required to work on assignments at least once every four weeks
Quiz	✓	✓	✓	✓	15%	
Final exam	✓	✓	✓	✓	60%	
Examination^: 60% (duration: 2 hours)						
** The weightings should add up to 100%.*					100%	

^ For a student to pass the course, at least 40% of the maximum mark for the continuous assessment and 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	CAPACITY for DIRECTED LEARNING to understand the concepts and implementation of key data structures and algorithms	High	Significant	Moderate	Basic	Not even reaching marginal levels
2. Mid-term and Final exams	ABIBILTY to apply the knowledge about the data structures and algorithms taught in the lectures and tutorials	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.)

Program correctness. Complexities of programs: notation, average and worst case analysis, complexities of common programming constructs. Sorting algorithms: merge sort, heap sort, quicksort, bucket sort. Algorithms for order statistics. Abstract data types: stacks, queues, heaps. Balanced search trees: AVL trees, red-black trees, 2-3 trees, B-trees. Hash tables. Merge-find sets.

Syllabus:

1. Program correctness and complexities
   Techniques for proving program correctness, e.g., loop invariant and induction. Asymptotic notations for program complexities. Summation and recurrence formulas. Complexities of common programming constructs, e.g., loops and recursive programs. Average and worst case analysis.

2. Sorting algorithms
   Selected sorting algorithms, such as merge sort, heap sort, quicksort, bucket sort, radix sort, as examples to illustrate the previous concepts and analysis techniques. Algorithms for order statistics.

3. Review of abstract data types
   Principles of abstract data types. Examples: stacks, queues, heaps.

4. Search trees
   Binary search trees. Balanced search trees: AVL trees, red-black trees, 2-3 trees, B-trees.

5. Hash tables
   Direct addressing. Hash functions. Collision resolution.

6. Merge-find sets
   Disjoint set operations. Path compression. Ackermann's function.

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.	Cormen T., Leiserson C., Rivest R. and Stein C. (2009). *Introduction to Algorithms.* MIT Press, 3rd edition

2.2 Additional Readings

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

1.	J. Lewis, J. Chase (2013). *Java Software Structures: Designing and Using Data Structures.* Pearson, 4th edition.
2.	Y. D. Liang (2013). *Introduction to Java™ Programming Comprehensive Version.* Pearson, 9th edition.