Xi'an Jiaotong-Liverpool University
西交利物浦大学

Department of Computing

MODULE HANDBOOK

CPT102

Data Structures

Steven Guan

Semester 2

2023/2024

SECTION A: Basic Information

- **Brief Introduction to the Module**

This module introduces students to a wide variety of data structures and algorithms. It provides students with a coherent knowledge of techniques for implementing data structures and algorithms and discusses the advantages and disadvantages of different data structures and algorithms. It also introduces students to those algorithms for fundamental tasks upon data structures, such as sorting and searching. Students will be encouraged to reflect upon the purpose of knowledge creation including their own motivations for engaging with data structure related research.

- **Key Module Information**

Module name: *Data Structures*
Module code: ***CPT102***
Credit value: *5*
Semester in which the module is taught: *S2*
Pre-requisites needed for the module:

Programmes on which the module is shared: *BEng Digital Media Technology, BSc Information and Computing Science*

- **Delivery Schedule**

Seminar room: *TBA*
Seminar time: *TBA*

- **Module Leader and Contact Details**

Name: *Prof. Steven Guan*

Brief Biography: *Steven Guan received his M.Sc. & Ph.D. from the University of North Carolina at Chapel Hill. He is currently a Professor and the Director for Research Institute of Big Data Analytics at Xi'an Jiaotong-Liverpool University (XJTLU). He served the head of department position at XJTLU for 4.5 years. Before joining XJTLU, he was a tenured professor and chair in intelligent systems at Brunel University, UK. Prof. Guan has worked in a prestigious R&D organization for several years, serving as a design engineer, project leader, and department manager. After leaving the industry, he joined later the Electrical & Computer Engineering Department at National University of Singapore as an associate professor.*

Prof. Guan's research interests include: machine learning, modelling, security, networking, and pseudorandom number generation. He has published extensively in these areas, with 140+ journal papers and 190+ book chapters or conference papers. He has chaired and delivered keynote speech for 100+ international conferences and served in 190+ international conference committees and 20+ editorial boards.

Email address: *steven.guan@xjtlu.edu.cn*
Office telephone number: *0512-8816-1501*
Room number and office hours: *SD425; by appointment*
Preferred means of contact: *e-mail*

Name: *Kok Hoe WONG*
Brief Biography of Kok Hoe Wong: *Upon completion of his Ph.D. in 3-D imaging, Dr. Kok Hoe WONG has worked for several renowned Multi-National Corporations (MNCs) before embarking into academia the last 10 years. His forte is in software engineering, project management, teaching and academic management. He has extensive experiences in architecting and managing enterprise-level IT projects, working with stakeholders from different parts of the world. Upon his arrival in China in 2007, he has progressed from being a Senior Lecturer to Vice President at a local institution, overseeing a successful Sino-Foreign partnership with Staffordshire University, UK. In recognition of his contributions, he had received several awards from the SuZhou government that include the "SIP Education Talent Award".*

Email address: *kh.wong@xjtlu.edu.cn*

Office telephone number: *8188-4951*

Room number and office hours: *SD431, by appointment*

Preferred means of contact: *e-mail*

SECTION B: What you can expect from the module

- **Educational Aims of the Module**

This module introduces students to a wide variety of data structures and algorithms. It provides students with a coherent knowledge of techniques for implementing data structures and algorithms

and discusses the advantages and disadvantages of different data structures and algorithms. It also introduces students to those algorithms for fundamental tasks upon data structures, such as sorting and searching. The student will be expected to solve data structure related problems using the knowledge and skills learnt and developed from the study of this module.

- **Learning Outcomes**

	LEARNING OUTCOMES
1	*Understand and to be able to apply a wide variety of data structures, together with their internal representation and algorithms;*
2	*Be able to make informed choices between alternative ways of implementation, justifying choices on grounds such as time and space complexity;*
3	*Be able to select, with justification, appropriate data structures to ensure efficient implementation of an algorithm.*

- **Assessment Details**

Initial Assessment

Sequence	**Method**	**Assessment Type (EXAM or CW)**	**Learning Outcomes Assessed** *(use codes under Learning Outcomes)*	**Duration**	**Week**	**% of Final Mark**	**Resit (Y/N/S)**
1	*Assessment Task 1*	*CW*	*ALL*			*10*	*S*
2	*Assessment Task 2*	*CW*	*ALL*			*10*	*S*
3	*Final Exam*	*EXAM*	*ALL*	*2 hours*		*80*	*S*

Resit Assessment

Assessment Type (EXAM or CW)	**Learning Outcomes Assessed** *(use codes under Learning Outcomes)*	**Duration**	**Week**	**% of Final Mark**
EXAM	*ALL*	*2 hours*		*100*

ASSESSED COURSEWORKS (1&2): (10% of the module mark each).

The task: *Quizzes will be conducted to examine the techniques learnt from the Data Structures module, your abilities to design & implement relevant data structures.*

Deadline: *The date for Quizzes will be announced at the beginning of the semester, typically scheduled near the latter part of the semester. Each student needs to attend Quiz under LM on the scheduled date.*

Grading: *Quizzes will be marked via LM/Quiz automatically with some feedback generated.*

- **Methods of Learning and Teaching**

Students will be expected to attend three hours of formal lectures in a typical week as well as to participate in pre-scheduled, supervised help sessions in a computer lab. Lectures will introduce students to the academic content and practical skills which are the subject of the module, while help sessions will allow students to interact with teaching assistants, develop and practice those skills relevant to coursework solving. In addition, students will be expected to devote three hours of unsupervised time to accomplish continuous assessment tasks and private study. Continuous assessment will be used to test to what extent practical skills have been learnt, in particular, assessment tasks will be solved individually and each solution comprises the resolution, based on Java and sound data structure techniques.

- **Syllabus & Teaching Plan**

Week number and/or date	***Lecture/Seminar/ Field trip/other***	***Topic/Theme/Title***	***Pre-reading (ch. refer to Required Textbook)***
Week 1	*Lectures*	*Introduction to Data Structures Data Structures, Data Types, and Abstraction Data Types;*	*Ch. 1.1, 1.2, 2.1*
Week 2	*Lectures*	*Abstraction, Information Hiding, and Encapsulation; Huffman Codes and ADT, Priority Queues; Efficiency; Static vs. Dynamic Data Structures*	*Ch. 1.1, 1.2, 2.1*
Week 3	*Lectures*	*Overview of Data Structure Programming, Programming with Java Libraries, Java Collections, Programming with Lists of Objects, Linear Collections vs Hierarchical Collections, Type Parameter, Abstract Datatype definition, Java arrays and ArrayLists; Collections and List, Using List and ArrayList, Iterating through List, Iterators, Interfaces vs Classes*	*Ch. 3.3*
Week 4	*Lectures*	*Bags, Sets, Queues, Lists, Stacks, Maps. Operations upon various data structures, Applications of data structures, Recursive methods, Stack ADT, Stack for web document processing, Program Stacks, Stack for postfix conversion, Stack for postfix processing, Stack for evaluating expressions, Java Stack Interface Specification, Examples of using Map, Iterating through a Map*	*Ch. 3.4, 3.6, 4.8*
Week 5	*Lectures*	*Queues and Priority Queues, Iterator vs Iterable; Sorting collections, Iterator vs Comparator, Comparator vs Comparable; Sorting with Comparators, Using Multiple Comparators, compareTo vs compare*	*Ch. 3.4, 3.7*
Week 6	*Lectures*	*Exceptions, Types of Exceptions, Catching exceptions, Throwing Exceptions, Implementing Collections, Abstract Classes, Interfaces vs Abstract Classes vs Classes, Lists Abstract Datatype definition; Defining ArrayList, Realisation of lists using arrays; Singly linked lists in Java (declaration, initialisation, use, iteration); Data representation of singly linked lists; Immutable List; ArrayList: fields and constructor, ArrayList methods; ArrayList: ensureCapacity; Cost of ArrayList operations, Cost of ArraySet operations, Time vs Space, Algorithm complexity*	*Ch. 3.4, 2.2, 2.3, 2.4*

Week 8	*Lectures*	*Recursions, recursion vs iteration; Testing collection implementations; Motivation for linked lists, Linked Structures, Linked structures for implementing Collections; linked lists operations, Memory allocation, Heap & memory allocation; Linked Node class, List using linked nodes with header; Using Linked Nodes, Creating & Iterating through a linked list; cost of Linked Collection methods*	*Ch. 1.3, 3.5.*
Week 9	*Lectures*	*Stacks/queues and their implementation; LIFO, Creating a Stack using a Linked List with a header; FIFO, Creating a Queue using a Linked List with a header; Application of Queues, User job queu e, Print spooling queue, I/O event queue*	*Ch. 3.6, 3.7, 6.2*
Week 10	*Lectures*	*Cost of ArraySet operations; Binary Search, Cost of SortedArraySet with Binary Search; Selection sort; Insertion sort; Merge sort; Quick sort; Bucket sort; slow sorts vs fast sorts; Sorting by Divide and Conquer; Cost of various sorting algorithms*	*Ch. 4.3, 7.1, 7.2, 7.3, 7.6, 7.7, 7.8, 2.3*
Week 11	*Lectures*	*Trees Abstract Datatype definition (trees, binary trees); Realisation of trees using references or arrays; Tree Ordering, Tree traversal, Breadth-First traversal; Tree and Recursion, Recursion tree;*	*Ch. 4.1, 4.2, 4.3, 4.4*
Week 12	*Lectures*	*Search lists, (binary) Search trees; Balanced Search Trees, AVL trees, AVL Rotation, AVL height balancing; Tree examples & applications; Tree implementation, Implementing Binary Trees, Implementing General Trees, Hash tables, Hash function, Keys, Security*	*Ch. 4.1, 4.2, 4.3, 4.4, 4.6, 4.8, 5.1-5.4, 5.6*
Week 13	*Lectures*	*Basic definitions of graph theory; Properties of graphs; Paths; Trees; Digraphs and their applications; network flows; Connected graphs; Incidence matrix and adjacency matrix of a graph; Trees and forests; Spanning trees, Minimum spanning tree, Greedy algorithm for determining a minimum spanning tree; Shortest path problem; Networks; To determine the maximum flow between two points (source and sink) in a network*	*Ch. 9.1, 9.4, 9.5*

- **Reading Materials**

Required (Essential) Textbook:

Title	Author	ISBN/Publisher
DATA STRUCTURES AND PROBLEM SOLVING USING JAVA	*M. A. WEISS*	*ADDISONWESLEY*

Recommended Texts:

Title	Author	ISBN/Publisher
DATA STRUCTURES AND ALGORITHMS IN JAVA	*M.T. Goodrich and R. Tamassia*	*PRENTICE HALL*

Additional Readings:

In addition to the required and recommended readings given above, you are encouraged to identify appropriate further resources for your study, e.g. articles in data structure related academic journals

SECTION C: Further Information

- **Student Feedback**
  The University requires student feedback to be obtained and evaluated by Departments for each module in every session. It is University policy that the preferred way of achieving this is by means of an Online Module Evaluation Questionnaire Survey. Students will be invited to complete the questionnaire survey for this module at the end of the semester.

You are strongly suggested to read policies mentioned below very carefully, which will help you better perform in your academic studies.
All the policies and regulations related to your academic study can be found in Student Academic Services section under the heading "Policies and Regulations" on eBridge.

- **Plagiarism, Cheating, and Fabrication of Data.**
  Offences of this type can result in attendance at a University-level committee and penalties being imposed. You need to be familiar with the rules. Please see the "Policy for Dealing with Plagiarism, Collusion and Data Fabrication" document available on e-Bridge in the Student Academic Services section under the heading 'Policies and Regulations'.

- **Rules of submission for assessed coursework**
  The School has detailed rules and procedures governing the submission of assessed coursework. You need to be familiar with them. Details can be found in the "Code of Practice for

Assessment" document available on e-Bridge in the Student Academic Services section under the heading 'Policies and Regulations'.

- **Late Submission of Assessed Coursework**
  The University attaches penalties to the late submission of assessed coursework. You need to be familiar with the University's rules. Details can be found in the "Code of Practice for Assessment" document available on e-Bridge in the Student Academic Services section under the heading 'Policies and Regulations'.

- **Mitigating Circumstances**
  The University is able to take into account mitigating circumstances such as illness or personal circumstances which may have adversely affected student performance on a module. It is the student's responsibility to keep their Academic Adviser, Programme Director or Head of Department informed of illness and other factors affecting their progress during the year and especially during the examination period. Students who believe that their performance on a examination or assessed coursework may have been impaired by illness, or other exceptional circumstances should follow the procedures set out in the Mitigating Circumstances Policy, which can be found on e-Bridge in the Student Academic Services section under the heading 'Policies and Regulations'.

- Learning Mall
  Copies of lecture notes and other materials are available electronically through Learning Mall, the University's virtual learning environment.

- *Attendance: Students who are able to be on campus are reminded of the Academic Policy requiring no less than 80% attendance at classes. Failure to observe this requirement may lead to failure or exclusion from resit examinations or retake examinations in the following year.*