Data Structures

CS 280 / SEM4504

VERSION FOR TRIMESTER 2, AY 2020/21

This module profile may be subject to change during the trimester – the online version is the authoritative version.

1. General Module Information

1.1 General Information

Credit*	5 credits
Pre-requisite(s) needed* (module code and name)	CS 225 / SEM2503 Advanced C/C++
Pre-requisite(s) for other modules* (module code and name)	CS 330 Algorithms Analysis
Co-requisite(s)* (module code and name)	NA
Module Owner (Institution)*	SIT
Module Owner (Programme)*	RTIS/IMGD/SEEMS
Cross Listing* (module code and name)	SEM4504 Data Structures
Result Type*	Grade
Module Coordinator	Name: Nisha Jain
	Phone:
	Email: nisha.jain@singaporetech.edu.sg
Module Co-Coordinator	Name: NA
	Phone:
	Email:
First taught in*	AY2020/21 Trimester 1

1.2 Module Descriptor*

The objective of this course is mainly to introduce the classical Abstract Data Types (ADTs) in Computer Science. The ADTs provide the hierarchical views of data organization used in programming. Fundamental data structures and their associated algorithms as well as complexity notation are introduced. Simply reading about data structures and algorithms and listening to a lecture is insufficient to master and implement these fundamental concepts. Every non-trivial program you write at DigiPen and in the real world will make heavy use of data structures and algorithms and this course enables you to reason about and apply them. CS225 is a prerequisite for this course.

1.3 Module Changes in Response to Previous Feedback:

NA

2. Module Learning Outcomes*

After successfully completing this module you should be able to:

1. Explain data structures, their associated algorithms, and their typical applications. Primary data structures that students will work with are arrays, lists, stacks, queues, trees, hash tables, and graphs.
2. Evaluate and differentiate data structure algorithms and implementations using algorithm complexity analysis techniques.
3. Implement several primary data structures (lists, trees, graphs, hash tables) as Abstract Data Types (ADTs).

3. Module Content and Learning Activities

The following lists the topics and learning activities that are planned for this module and the amount of study hours associated with them.

3.1 Topics covered in the module

Topic	**Details**	**Contact hours** (optional)
Overview	Motivation for data structures and algorithms analysis, basic code design	1
Memory Management	Definitions and properties, operations, identities, equivalence relations	3

Basic Algorithms Analysis	Big-Oh notation, Simplified Model of Computing, asymptotic behaviour, common complexity classes	5
Sorting	Bubble sort, insertion sort, selection sort, merge sort, quick sort,.	2
Recursion	Recursion fundamentals, tail recursion, recurrence relations for complexity analysis	2
Abstract Data Types (ADTs)	ADT fundamentals, stacks, queues, postfix notation	4
Binary Trees	Terminologies, definitions, traversals, binary search trees (BSTs), AVL trees, Splay trees	9
Graphs	Terminologies, definitions, traversals, Dijkstra's algorithm, minimum spanning trees (MSTs)	9
Heaps	Heaps and heapsort	4
Hashing	Hashing fundamentals, hash functions, collision resolution, separate chaining, open addressing,	5
B-Trees	2-3 trees, 2-3-4 trees, red-black trees	4
Total:		48

3.2 Learning Activities

Type of Activity	Details	Total study load [hrs]
Lectorials	Lectures with interactive quizzes to reinforce and deepen understanding of the module material.	48
Programming Assignment Work	Students will independently work on around 6 programming assignments to translate the knowledge acquired into actual working code to address carefully crafted software requirements.	60
Programming Assignment Consultations	Students will interact with instructors and student mentors in a mix of face-to-face and online sessions to obtain supervision on the programming assignments.	18
Total:		126

External Preparation:

It is expected that the students in this class spend ___ hours on average per week for outside classroom activities through the trimester, including, but not limited to, homework, reading assignments, project implementation, group discussions, preparation of examinations, etc.

A tentative schedule is listed below, and it is subject to change. Please follow the module details on the LMS for any updates.

Week	Activity Type	Topics	Duration [hrs]
1	Lectorial	Overview & Memory Management	4
	Programming Assignment Consultations	Object Allocator	1.5
2	Lectorial	Basic Algorithms Analysis	4
	Programming Assignment Consultations	Object Allocator	1.5
3	Lectorial	Sorting & Recursion	4

Week	Activity Type	Topics	Duration [hrs]
	Programming Assignment Consultations	Object Allocator	1.5
4	Lectorial	ADTs	4
	Programming Assignment Consultations	B-Lists	1.5
5	Lectorial	BSTs	4
	Programming Assignment Consultations	B-Lists	1.5
6	Lectorial	AVL & Splay Trees	4
	Programming Assignment Consultations	Trees	1.5
7		Break	
8	Lectorial	Graphs I	4
	Programming Assignment Consultations	Trees	1.5
9	Lectorial	Graphs II	4
	Programming Assignment Consultations	Graphs	1.5
10	Lectorial	Heaps	4
	Programming Assignment Consultations	Graphs	1.5
11	Lectorial	Hash Tables	4
	Programming Assignment Consultations	Hash Tables	1.5
12	Lectorial	B-trees	4
	Programming Assignment Consultations	Hash Tables	1.5
13	Lectorial	Revision	4

Week	Activity Type	Topics	Duration [hrs]
	Programming Assignment Consultations	Sudoku (bonus)	1.5

4. Learning Resources

4.1 Required Resources

NA

4.2 Recommended Resources

- *Algorithms in C++, Parts 1-4: Fundamentals, Data Structure, Sorting, Searching, Third Edition, by Robert Sedgewick. Copyright © 1998 by Addison-Wesley Publishing Company, Inc. (ISBN: 0-201-35088-2).*
- *Introduction to Algorithms, Third Edition, Cormen et al. Copyright © 2001 by MIT. (ISBN: 0-262-03384-4).*
- *Data Structures and Algorithms with Object-Oriented Design Patterns in C++, Copyright © 1998 by John Wiley & Sons (ISBN: 0471-24134-2). E-book also available at http://www.brpreiss.com/books/opus4/.*

4.3 Any Other Learning Resources

NA

5. Assessment

5.1 Assessment Tasks*

This is a summary of the assessment in the module. For detailed information on each assessment, refer to the LMS assignment descriptions. Dates for assessments are subjected to minor amendment and early notification will be made should changes be required.

Assessment Task	Weighting	Tentative week/due date

Programming Assignments Students individually apply techniques taught in lectorials to complete working software to address requirements in assignment problems.	35%	Week 1-13
Quizzes Students individually take short written tests to revise and check-in on the concepts learnt.	15%	Week 2-12
Midterm Test Students individually take a longer written test to perform a midterm check on their learning using applied-focused questions.	20%	Week 8
Final Test Students individually take a longer written test to perform a final check on their learning using applied-focused questions.	30%	Week 14/15

5.2 Pass Requirements

An “E” grade or above is considered as “Pass” and an “F” grade is considered as “Fail”. Students with grades “D”, “E” or “F” will be given the option to repeat the module (i.e., re-module) but the maximum GPA for the repeated attempt will be capped at 2.0. Successful completion of the academic module is also based on fulfilment of any other requirement(s) set by the module coordinator and programme director.

Marks are final after endorsement by the Board of Examiners.

Do note that the criteria for acceptable standing in any given trimester is maintaining a minimum Cumulative Grade Point Average (CGPA) of 2.0. Refer to the SIT Academic Guide for further details.

5.3 Late Submission

Programming Assignments:

The due day/time will be published on the class website when each project is assigned. Late submissions are allowed with the following penalty:

- Within two days after the due day (-20%)
- Assignments turned in more than two days late will not receive credit

5.4 **Disability Support Services**:

Students who have special needs or medical conditions and require formal accommodations in order to fully participate or effectively demonstrate learning in this class should contact the Student Life & Advising Office (studentlife.sg@digipen.edu) at the beginning of each semester. A Student Life & Advising Officer will meet with the student privately to discuss how the accommodations will be implemented.