COURSE OUTLINE: MH1403

Course Title	**Algorithms and Computing**		
Course Code	**MH1403**		
Offered	Study Year 1, Semester 2		
Course Coordinator	Wu Hongjun (Assoc Prof)	wuhj@ntu.edu.sg	6513 7192
Pre-requisites	PS0001 OR BS1009 OR CV1014 OR MS1008 OR MA1008 OR {CB0494, CH2107} OR {CB0494, BG2211}		
Co-requisites	CB0494		
AU	3		
Contact hours	Lectures: 26, Laboratories: 8, Tutorials: 8		
Approved for delivery from	AY 2022/23 semester 2		
Last revised	3 Jan 2023, 13:48		

Course Aims

This course aims to give you a systematic introduction to data structures and algorithms for constructing efficient computer programs. Emphasis is on data abstraction issues in the program development process, and on the design of efficient algorithms. Simple algorithmic paradigms such as greedy algorithms, divide-and-conquer algorithms and dynamic programming will be introduced. Elementary analyses of algorithmic complexities will also be taught.

Intended Learning Outcomes

Upon successfully completing this course, you should be able to:

1. Implement data structures in Python.
2. Analyze the complexity of algorithms.
3. Design and implement efficient algorithms for given applications.
4. Solve large problems systematically and effectively.
5. Work as a team and collaborate to manage larger programming projects.

Course Content

Introduction to the course

Algorithm Analysis

Python Classes

Array, Linked List, Python List

Stack

Queue

Tree

Binary Search Tree

AVL Tree

Sorting Algorithms

Greedy algorithms

Divide-and-Conquer algorithms

Dynamic programming

Assessment

Component	Course ILOs tested	SPMS-MAS Graduate Attributes tested	Weighting	Team / Individual	Assessment Rubrics
Continuous Assessment					
Laboratories					
Assignment	1, 2, 4, 5	**1.** c, d **2.** a, b, c **4.** a **5.** a	16	individual	See Appendix for rubric
Mid-semester Quiz					
Short Answer Questions	1, 2, 3, 4	**1.** c, d **2.** a, b, c **4.** a **5.** a	24	individual	See Appendix for rubric
Examination (2 hours)					
Short Answer Questions	1, 2, 3, 4	**1.** c, d **2.** a, b, c **4.** a **5.** a	60	individual	See Appendix for rubric
		Total	**100%**		

These are the relevant SPMS-MAS Graduate Attributes.

1. Competence

c. Discover patterns by abstraction from examples

d. Use computer technology to solve problems, and to communicate mathematical ideas

2. Creativity

a. Critically assess the applicability of mathematical tools in the workplace

b. Build on the connection between subfields of mathematics to tackle new problems

c. Develop new applications of existing techniques

4. Civic-mindedness

a. Develop and communicate mathematical ideas and concepts relevant in everyday life for the benefits of society

5. Character

a. Act in socially responsible and ethical ways in line with the societal expectations of a mathematics professional, particularly in relation to analysis of data, computer security, numerical computations and algorithms

Formative Feedback

You will be receiving formative feedback verbally during tutorials and through written grading of your lab assignments. You will receive summative group feedback on the exam following the conclusion of the course.

Learning and Teaching Approach

Lectures (26 hours)	Lectures are to introduce the basic knowledge on data structure, algorithms and complexity analysis. The concept of each lecture will be enhanced by the tutorial and lab sessions after the lectures.
Laboratories (8 hours)	Lab sessions will be on the format of both individual and group projects, you are expected to solve the given project by Python programming languages. Lab tutors will be present at the lab sessions only to guide you in finding solutions by yourselves
Tutorials (8 hours)	Tutorials are divided into two parts: introduction to the programming of the data structures, and solution solving. You are expected to solve the tutorial questions by yourselves before coming to the tutorial sessions, and solutions will only be provided after that.

Reading and References

- Textbook: Michael T. Goodrich and Roberto Tamassia, Algorithm Design and Applications, WILEY, 2014 (978-1-118-33591-8)
- Reference: Huang Guangbin and Ng Jim Mee, Data structures and algorithms, Pearson Education South Asia, 2007 (9789810679149)
Both books will be available at NTU libraries.

Course Policies and Student Responsibilities

You are expected to attend the lectures and the tutorial sessions, and to take all scheduled assignments and projects by due dates. Not submitting a lab assignment or project before the corresponding deadline will be counted as no submission. You are expected to take responsibility to follow up with course notes, assignments and course related announcements they have missed.

Academic Integrity

Good academic work depends on honesty and ethical behaviour. The quality of your work as a student relies on adhering to the principles of academic integrity and to the NTU Honour Code, a set of values shared by the whole university community. Truth, Trust and Justice are at the core of NTU's shared values.

As a student, it is important that you recognize your responsibilities in understanding and applying the principles of academic integrity in all the work you do at NTU. Not knowing what is involved in maintaining academic integrity does not excuse academic dishonesty. You need to actively equip yourself with strategies to avoid all forms of academic dishonesty, including plagiarism, academic fraud, collusion and cheating. If you are uncertain of the definitions of any of these terms, you should go to the Academic Integrity website for more information. Consult your instructor(s) if you need any clarification about the requirements of academic integrity in the course.

Course Instructors

Instructor	Office Location	Phone	Email
Wu Hongjun (Assoc Prof)	SPMS-MAS-05-47	6513 7192	wuhj@ntu.edu.sg

Planned Weekly Schedule

Week	Topic	Course ILO	Readings/ Activities
1	Introduction to the course, Algorithm Analysis	1, 2, 3, 4	Lecture notes
2	Python Classes	1, 2, 3, 4	Lecture notes
3	Array, Linked List, Python List	1, 2, 3, 4	Lecture notes
4	Stack and Queue	1, 2, 3, 4	Lecture notes
5	Tree	1, 2, 3, 4	Lecture notes
6	Binary Search Tree	1, 2, 3, 4	Lecture notes
7	AVL Tree	1, 2, 3, 4	Lecture notes
8	Sorting Algorithms	1, 2, 3, 4	Lecture notes
9	Divide-and-Conquer Algorithms	1, 2, 3, 4, 5	Lecture notes
10	Midterm Test	1, 2, 3, 4, 5	Midterm test
11	Greedy Algorithms	1, 2, 3, 4, 5	Lecture notes
12	Dynamic Programming	1, 2, 3, 4, 5	Lecture notes
13	Introduction to Linear programming, Integer programming, and Revision	1, 2, 3, 4	Lecture notes

Appendix 1: Assessment Rubrics

Rubric for Laboratories: Assignment (16%)

The students submit the lab programming codes. Each lab submission is four marks. The guideline for marking each code of the lab submission,

code works well and the algorithm in the code is good	full marks
code is partially correct (the code works for some cases)	30% to 90% of the full marks depending on how many cases the code covers
code gives correct outputs, but the code is very inefficient	50% to 80% of the full marks
code does not work, the student knows how to solve the problem (the student cannot debug properly)	50% to 80% of the full marks
code does not work, and the student does not know how to solve the problem	zero marks

Rubric for Mid-semester Quiz: Short Answer Questions (24%)

Point-based marking (not rubric-based). Short answer questions.

Rubric for Examination: Short Answer Questions (60%)

Point-based marking (not rubric-based). Short answer questions.