COURSE SYLLABUS

CSC10004 – Data Structures and Algorithms

1. GENERAL INFORMATION

Course name:	Data structures and Algorithms
Course name (in Vietnamese):	Cấu trúc dữ liệu và Giải thuật
Course ID:	CSC10004
Knowledge block:	
Number of credits:	4
Credit hours for theory:	45
Credit hours for practice:	30
Credit hours for self-study:	90
Prerequisite:	
Prior-course:	Programming Techniques
Instructors:	

2. COURSE DESCRIPTION

The course is designed to explore the fundamental algorithms and data structures which lay the cornerstone for most studies in computer science. Students will first learn the basic algorithms for search and sorting, reflecting the everyday tasks in practice. Next, students will experience various abstract data structures, including linked lists, hash tables, trees, and graphs, to learn how to implement them in several ways and how the choice affects their efficiencies. Since practical problems are usually resource-limited, it is essential to analyze the strategies in terms of time and space complexities using asymptotic notations, leading to appropriate algorithms and data structures. Students are expected to develop their algorithmic thinking and enhance their programming skills throughout the course.

3. COURSE GOALS

At the end of the course, students are able to

ID	Description	Program LOs
G1	Interpret the fundamental concepts and terminologies of Data structures and Algorithms in the given context	1.2, 1.3.1
G2	Analyze the algorithms and data structures in terms of time and space complexities using asymptotic notations	1.3.1, 4.1.2, 4.1.4
G3	Choose the algorithms and data structures following the problem requirements (e.g., data size and distribution, a tradeoff between execution time and storage)	4.1.1, 4.1.2, 4.1.3, 4.3.1, 4.3.2, 4.3.3
G4	Implement the algorithms and data structures using C/C++ programming language	1.3.1, 5.3.2
G5	Promote personal aptitudes for logical thinking and communication	2.1.1, 2.1.2, 2.1.5, 2.3.1, 2.3.2, 2.4, 4.1.3, 4.3

4. COURSE OUTCOMES

CO	Description	I/T/U
G1.1	Understand the fundamental concepts and terminologies of Data structures and Algorithms, and apply them in the proper context	I/T
G1.2	Understand textbooks and tutorials about Data structures and Algorithms	U
G2.1	Explain the asymptotic notations: Big-O, Big-Ω, and Big-Θ	T
G2.2	Show the Big-O time complexity of basic algorithms and data structures	T
G3.1	Illustrate the basic search and sorting algorithms	T
G3.2	Compare the search and sorting algorithms on execution time and storage	T/U
G3.3	Identify the properties and applications of each abstract data structure	T
G3.4	Compare the abstract data structures on execution time and storage	T/U
G3.5	Manipulate the algorithms on graphs, including graph traversals, topology sorting, finidng minimum spanning tree, and finding shortest paths	T

G4.1	Implement the search and sorting algorithms using C/C++	T/U
G4.2	Implement the abstract data structures using different primitives structures	T/U
G5.1	Develop practical personal communication skills, both oral and writing	U
G5.2	Develop the personal aptitudes for logical thinking	U

5. TEACHING PLAN

ID	Topic	Course outcomes	Teaching/Learning Activities	Assessments
1	Introduction to DSA	G1.1-2	Lecturing	
2	Algorithm efficiency: Big-O, Big-Ω, and Big-Θ	G1.1-2 G2.1-2	Lecturing and Case Studies	Q1
3	– Search algorithms: Linear search and Binary search – Sorting algorithms: $O(n^2)$ algorithms ▪ Required: Selection Sort and Insertion Sort ▪ Optional: Bubble Sort	G1.1-2 G2.2 G3.1-2	Lecturing and Case Studies	Q2
4	– Sorting algorithms: $O(n\log_2 n)$ algorithms ▪ Required: Heap Sort and Quick Sort ▪ Optional: Merge Sort – Sorting algorithms: O(n) algorithms (Radix Sort)	G2.2 G3.1-2	Lecturing and Case Studies	Q3
5	Reviews: Linked lists, Stack, Queue and Priority Queue	G1.1-2 G3.3-4	Lecturing and Case Studies	Q4
6	Trees: Binary search tree	G1.1-2 G3.3-4	Lecturing and Case Studies	Q5
7	**Midterm Examination** Trees: AVL tree	G2.2 G3.3-4	Lecturing and Case Studies	Q6

8	Trees: B-tree with optional extensions covering Red-black, AA, 2-3, 2-3-4 tree	G2.2 G3.3-4	Lecturing and Case Studies	Q7
9	Graphs: Basic concepts and graph traversals	G1.1-2 G3.5	Lecturing and Case Studies	Q8 **GA**
10	Graphs: Topological sort, Minimum spanning tree, and Shortest path	G3.5	Lecturing and Case Studies	Q9
11	ADT: Hash tables and collision resolution ▪ Required: Linear Probing and Chaining ▪ Optional: quadratic probing and double hashing	G1.1-2 G3.3-4	Lecturing and Case Studies	Q10

6. LABORATORY WORKING PLAN

The teaching assistant and lab instructors are responsible for

- Consolidating the students' understands on Data structures and Algorithms and their programming skills, and
- Giving, correcting, and grading assignments.

Students will have weekly classes for laboratory work.

Furthermore, the teaching assistant is available for Q&A at his/her office hours.

ID	Topic	Course outcomes	Teaching/Learning Activities	Assessments
1	Revision on programming skills		Self-study: C/C++ programming	
2	Revision on programming skills		Self-study: C/C++ programming	
3	Algorithm efficiency: Big-O, Big-Θ, and Big-Ω	G2.1-2	Code tutorials and weekly lab assignment	

4	Search algorithms $O(n^2)$ sorting algorithms	G3.1-2 G4.1	Code tutorials and weekly lab assignment	
5	$O(n\log_2 n)$ sorting algorithms	G3.1-2 G4.1	Code tutorials and weekly lab assignment	
6	O(n) sorting algorithms	G3.1-2 G4.1	Code tutorials and weekly lab assignment	
7	Hash tables	G3.3-4 G4.2	Code tutorials and weekly lab assignment	
8	Binary search tree	G3.3-4 G4.2	Code tutorials and weekly lab assignment	
9	Trees: AVL tree	G3.3-4 G4.2	Code tutorials and weekly lab assignment	
10	Basic graph concepts	G3.5 G4.2	Code tutorials and weekly lab assignment	
11	**Final Lab Exam**	G3.1-5 G4.1-2 G5.1-2		

7. ASSESSMENTS

ID	Topic	Description	Course outcomes	Ratio (%)
A1	**Coursework**			**10% - 20%**
A11	Quizzes	In-class assignments. Announced. **Closed-book.** They are on any topics in any lecture covered and any reading material assigned up to the time the quiz is administered.	G1.1-2 G2.1-2 G3.1-5	10%

A3	**Lab works**			**30% – 40%**
A31	Weekly Labs	**Code** mini C++ programs to implement data structures and algorithms given in the lectures	G2.1-2 G3.1-5 G4.1-2	20%
A32	Lab Final Exam	**90-minute** exam. Announced**. Closed-book**. **Code** C++ programs to solve **the** given problems by using any fundamental techniques in the course	G3.1-5 G4.1-2 G5.1-2	10%
A4	**Examinations**			**50% – 60%**
A41	Midterm Exam	**60-minute** exam. Announced. **Closed-book**. They are on any topics in any lecture covered and any reading material assigned up to the time the exam is administered	G1.1-2 G2.1-2 G3.1-5 G5.1-2	20%
A42	Final Exam	**90-minute** exam. Announced**. Closed-book**. They are on any topics in any lecture covered and any reading material assigned up to the time the exam is administered	G1.1-2 G2.1-2 G3.1-5 G5.1-2	40%
A5	**Bonus**			**Max 10%**
A51	Quizzes and Challenges	In-class assignments. They are on any topics in any lecture covered and any reading material assigned up to the time the quiz is administered.	G1.1-2 G2.1-2 G3.1-5	10%

8. RESOURCES

Textbooks

- Carrano, Frank M. and Henry, Timothy. 2014. **Data Abstraction & Problem Solving with C++: Walls and Mirrors**. Sixth Edition. Pearson.

- Cormen, T.H., Leiserson, C.E., Rivest, R.L. and Stein, C., 2009. **Introduction to algorithms**. Third Edition. MIT press.

Others

• Dương Anh Đức và Trần Hạnh Nhi. 2003. **Nhập môn Cấu trúc dữ liệu và giải thuật**. NXB ĐHQG TP.HCM.

9. GENERAL REGULATIONS & POLICIES

- All students are responsible for reading and following strictly the regulations and policies of the school and university.
- Students absent for more than three theory sessions are not allowed to take the exams.
- Students are encouraged to form study groups to discuss the topics. However, individual work must be done and submitted on your own.
- For any kind of cheating and plagiarism, students will be graded 0 for the course. The incident is then submitted to the school and university for further review.
- Students who are absent for mid-term exam or final exam and students who have lab works scores less than 10% are considered as unqualified for course completion.
- Programming assignments must be compiled with C++ compilers. No credit will be given for code that does not compile. This means that all components of a program must be compiled together, or students will not receive any credit for any of them.
- The following policies are specific for the class 23CLC03.
  - The lab instructors have the rights to specify the IDE and compiler for all the assignments and examinations in the lab works.
  - If the student plans to be absent from a lecture, he/she must complete the corresponding in-class assignment during the lab session before the lecture. Late requests will not be accepted.

Ho Chi Minh City, May 15th, 2025

DEAN OF FACULTY

Đinh Bá Tiến