

**1. Course Number and Course Title:**

CMP 305 – Data Structures and Algorithms

**2. Credit Hours:**

3 – 2 – 3

**3. Prerequisites and/or Co-Requisites:**

Prerequisite: CMP 220 Programming II

Prerequisite/concurrent: CMP 213 Discrete Structures or MTH 213 Discrete Mathematics

Co-requisites: CMP 305L

**4. Name and Contact Information of Instructor: Dr. Mohamed Hamza Ibrahim**

- Email: [meibrahim@aus.edu](mailto:meibrahim@aus.edu)
- Office: **ESB-2131**
- Phone: **(06) 5154828**
- Office Hours: **Posted on office door and iLearn**; also, by appointment via email

**5. Course Description (Catalog Description):**

Covers the design, analysis, and implementation of abstract data types and related algorithms to solve computing problems efficiently. Includes fundamental data structures such as arrays, linked lists, stacks, and queues, as well as advanced data structures such as trees, hash tables, heaps, and graphs. Studies algorithms for manipulating these data structures, recursive programming, searching, and sorting. Laboratory work includes substantial programming assignments.

**6. Textbook and other Supplemental Material:**

Textbook:

- M.A. Weiss, *Data Structures and Algorithms Analysis in C++*, 4th ed., Addison-Wesley/Pearson, 2014.

Supplemental material:

- T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, *Introduction to Algorithms*, 4th ed., MIT Press, 2022.
- A. Drozdek, *Data Structures and Algorithms in C++*, 4th ed., Cengage, 2013.
- D.S. Malik, *C++ Programming: Program Design Including Data Structures*, 8th ed., Cengage, 2017.

**7. Course Learning Outcomes:**

Upon completion of the course, students will be able to:

1. Derive the time complexity of basic programs and algorithms to evaluate their performance when handling large amounts of data.
2. Implement Abstract Data Types (ADTs) such as vectors, lists, stacks, queues, trees, heaps, priority queues, hash tables, and graphs.
3. Develop programs that make use of one or more ADTs, using the C++ object-oriented programming language and the C++ Standard Library.
4. Describe the use of recursion to process data structures and implement recursive algorithms.
5. Apply searching and sorting algorithms to solve computing problems effectively.

**8. Teaching and Learning Methodologies:**

Methods include lectures, problem-based learning, class discussions, and group work, as well as laboratory sessions. Students learning is assessed via in-class quizzes, exams, homework, and programming assignments.

**9. Course Topics and Schedule:**

Stage	Topic(s)	Week
Foundation	Introduction to data structures and algorithms Algorithm analysis, computational time complexity	Week 1
	Abstract data types, containers, C++/STL library (vector class, List class), and iterator basics	Week 2
Linear Structures	Linked lists, arrays, and performance comparison	Week 3
	Implementing List ADT using pointer-based (linked list) design	Week 4
	Stack and Queue ADTs and STL implementations	Week 5
Algorithmic Strategies	Recursion, design and programming	Week 6
	Accumulator recursion	Week 7
Hierarchical Structures	Trees, binary trees and tree traversals	Week 8
	Binary search trees, applications, and sorted list	Week 9
	Tree performance, balanced trees, and AVL trees	Week 10
Priority & Associative Structures	Heaps, array-based binary heaps, and priority queues	Week 11
	Hash tables, associative arrays, and hashing functions	Week 12
Non-Linear Graph Structures	Graph representations and traversal algorithms (BFS, DFS)	Week 13
Algorithmic Utilities	Sorting and searching strategies and algorithms	Week 14
	<b>Final Exam</b>	Week 15

**10. Schedule of Laboratory and other Non-Lecture Sessions:**

Assignment	Due Date (tentative)
Lab 1 – Review of C++ classes and operators	Week #2
Lab 2 – Algorithmic complexity	Week #3
Lab 3 – STL data structures and algorithms	Week #4
Lab 4 – Implementation of singly linked lists	Week #5
Lab 5 – Doubly-linked lists and applications	Week #6
Lab 6 – Coding vector, list, iterator classes	Week #7
Lab 7 – Application of stacks and queues	Week #8
Lab 8 – Review and maze application	Week #9

Lab 9 – Recursive programming, basics	Week #10
Lab 10 – Recursive programming, advanced	Week #11
Lab 11 – Trees and traversals, binary trees	Week #12
Lab 12 – Binary search trees, Balanced binary search trees, application	Week #13
Lab 13 – Hashing and hash table, and Graph and traversal algorithms	Week #14

**11. Out-of-Class Assignments with Due Dates:**

Assignment	Due Date (tentative)
Homework 1	Week #4
Homework 2	Week #6
Homework 3	Week #9

**12. Student Evaluation:**

Assessment	Weight	Due Date (tentative)
In-class Quizzes	6%	5 <sup>th</sup> , 7 <sup>th</sup> and 11 <sup>th</sup> weeks of classes
Homework Assignments	6 %	cf. section 11
Labs	8 %	cf. section 10
Midterm 1 Exam	20 %	Week #8
Midterm 2 Exam	20 %	Week #12
Final Exam	40 %	As per registrar’s schedule

**13. Assessment Instruments:**

Assessment	Course Learning Outcomes
In-class Quizzes	O1 – O5
Homework Assignments	O1 – O5
Laboratory Assignments	O2 – O5
Midterm 1 Exam	O1 – O3
Midterm 2 Exam	O2 – O4
Final Exam	O1 – O5

**14. Contribution of Course to Program Outcomes:**

BSCS Program Outcomes	Emphasis in this course	Course Learning Outcomes
(1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions.	●	O1–O5
(2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program’s discipline.	●	O2–O5
(3) Communicate effectively in a variety of professional contexts.		
(4) Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.		
(5) Function effectively as a member or leader of a team engaged in activities appropriate to the program’s discipline.	○	O3
(6) Apply computer science theory and software development fundamentals to produce computing-based solutions.	●	O2–O5

Emphasis: ● High; ◐ Medium; ○ Low; Blank – Nothing Specific Expected

BSCoE Program Outcomes	Emphasis in this course	Course Learning Outcomes
(1) Identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics	●	O2–O5
(2) Apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors	○	O3,O5
(3) Communicate effectively with a range of audiences		
(4) Recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts		
(5) Function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives	○	O3
(6) Develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions	○	O2–O5
(7) Acquire and apply new knowledge as needed, using appropriate learning strategies		

Emphasis: ● High; ◐ Medium; ○ Low; Blank – Nothing Specific Expected

#### 15. Letter Grade Policy:

Total (T)	Letter Grade
$95 \leq T$	A
$90 \leq T < 95$	A-
$85 \leq T < 90$	B+
$80 \leq T < 85$	B
$75 \leq T < 80$	B-
$70 \leq T < 75$	C+
$65 \leq T < 70$	C
$60 \leq T < 65$	C-
$55 \leq T < 60$	D
$T < 55$	F

#### 16. Course Policies:

- The course will follow selected subjects as listed on the course schedule.
- Course Website (iLearn, <https://ilearn.aus.edu/>). Students are asked to check the website regularly for announcements.

- There will be no make-up exams without a valid reason according to AUS rules. If a student misses an exam without a valid reason, then s/he will get a grade of 0.
- **Attendance:**
  - **Attendance is mandatory according to AUS rules** and will be taken using iLearn during the lecture.
  - **The AUS attendance policy of 15% maximum absences allowed will be enforced. Upon breaching the 15% absence rule (regardless the absence is justified or not), the student will be automatically dropped from the course with a grade of W.**
  - Attending labs is also a must. If a student misses a lab, then s/he will get a grade of 0 for that lab work.
- **Academic Integrity:**
  - AUS academic integrity code will be strictly enforced.
  - All submissions (homework, labs, quizzes, exams) within this course are individual submissions and the effort of a student registered within this course.
  - Any sort of communication with another individual or using any medium of help other than the allowed material (i.e. course textbook) is prohibited.
  - The use of any **Generative Artificial Intelligence** model (e.g., ChatGPT or similar) in attempting homework, labs, quizzes, or exams will be considered a violation of the AUS academic integrity code. **It is considered an academic integrity violation to represent the output of a generative artificial intelligence tool as your own work.**
  - In the unlikely event of academic dishonesty in coursework (apart from written exams or quizzes), a grade of zero is given in all coursework. Additionally, the incident will be reported to the dean's office for further disciplinary action.
  - Any academic misconduct and violation of the code will be treated seriously with penalties including a failing grade of F for the course, or suspension for the semester.
- **Makeup Policy for Missed Quizzes/Tests:** No makeup quizzes or tests will be offered if a student misses a midterm exam or a quiz without a valid reason according to AUS rules, the student will receive a grade of 0 for that exam/quiz.
- **Grading Policy:** Grades for assessments will be made available on iLearn after each assessment. If you have any concerns or disputes about your grades, please submit them within one week of the posting date.
- **Home assignments:** Once an assignment is posted, no extension will be given. Late submissions will not be accepted.
- **Grade curving:** There will be no curving up for the grades.