Computer Science (COMP) 272

Data Structures and Algorithms (Revision 7)

Status:	Replaced with new revision, see the **course listing** for the current revision
Delivery mode:	**Individualized study online**
Credits:	3
Area of study:	Science
Prerequisites:	**COMP 268** or **COMP 206**. Familiarity with the fundamentals of Java and/or C++ is a prerequisite to this course. Candidates with considerable programming skills in Java, C, C++, or other languages may be admitted upon approval from the course professor. Knowledge of high school mathematics (MATH 30 level) is assumed.
Precluded:	None
Challenge:	COMP 272 has a challenge for credit option.
Faculty:	**Faculty of Science and Technology**

Notes:	Students who are concerned about not meeting the prerequisites for this course are encouraged to contact the **course coordinator** before registering.

Overview

COMP 272 covers analysis and design of fundamental data structures and engages learners to use data structures as tools to algorithmically design efficient computer programs that will cope with the complexity of actual applications.

The course focuses on basic and essential topics in data structures, including array-based lists, linked lists, skiplists, hash tables, recursion, binary trees, scapegoat trees, red–black trees, heaps, sorting algorithms, graphs, and binary trie.

Outline

- Unit 0: Orientation
- Unit 1: Introduction
- Unit 2: Array-Based Lists
- Unit 3: Linked Lists
- Unit 4: Skiplists
- Unit 5: Hash Tables
- Unit 6: Recursion
- Unit 7: Binary Trees
- Unit 8: Scapegoat Trees

- Unit 9: Red–Black Trees
- Unit 10: Heaps
- Unit 11:Sorting Algorithms
- Unit 12: Graphs
- Unit 13: Binary Trie

Learning outcomes

Upon successful completion of this course, you should be able to

- explain the systematic methods of efficiently organizing and accessing data in data structures and algorithms.
- identify the properties and structural patterns in data structures.
- apply abstract data types to the design of data structures.
- analyze algorithms using a mathematical notation and experimental studies.
- perform comparative analysis of the typical data structures and algorithms.
- design and analyze recursive algorithms in data structures.
- write code in pseudocode and high-level programming languages for the implementation of various data structures and algorithms.

Evaluation

To **receive credit** for COMP 272, you must achieve a course composite grade of at least **D (50 percent)** and a grade of at least 50 percent on the final examination. The weighting of the composite grade is as follows:

Activity	Weight
Assignment 1 (Unit 1-4)	20%

Activity	Weight
Assignment 3 (Unit 9-13)	20%
Final Exam	40%
Total	**100%**

The **final examination** for this course must be requested in advance and written under the supervision of an AU-approved exam invigilator. Invigilators include either ProctorU or an approved in-person invigilation centre that can accommodate online exams. Students are responsible for payment of any invigilation fees. Students are responsible for payment of any invigilation fees. Information on exam request deadlines, invigilators, and other exam-related questions, can be found at the **Exams and grades** section of the Calendar.

To learn more about assignments and examinations, please refer to Athabasca University's **online Calendar** .

Materials

This course either does not have a course package or the textbooks are open-source material and available to students at no cost. This course has a **Course Administration and Technology Fee** , but students are not charged the Course Materials Fee.

The main text for this course is Pat Morin's online book titled *Open Data Structures*. You can choose to complete this course using either Java or using C++, or both. The main text can be **accessed here**. You may choose Java edition, C++ edition, or pseudo-code edition for the textbook at **opendatastructures.org** as well. (Online)

Other Materials

- Units 0 through 13 of the study guide
- descriptions of the requirements for the individual assignments
- sample exam

Additional supporting materials of interest to students may occasionally be made available electronically.

Special Course Features

Students are required to have access to the Internet.

The course work in COMP 272 requires students to have an appropriate programming environment or tool for Java or C++ programming in their local computer(s). More information about programming environment and tools needed to implement any assignment are detailed in the course package.

Challenge for credit

Overview

The challenge for credit process allows you to demonstrate that you have acquired a command of the general subject matter, knowledge, intellectual and/or other skills that would normally be found in a university-level course.

Full information about **challenge for credit** can be found in the Undergraduate Calendar.

Evaluation

To **receive credit** for the COMP 272 challenge registration, you must achieve a grade of at least 75 per cent on the assignment and **B (75 percent)** on the final examination. The weighting of the composite grade is as follows:

Activity	Weight
Assignment	50%
Final Examination	50%
Total	**100%**

Challenge for credit course registration form

Important links

- **Academic advising**
- **Program planning**
- **Request assistance**
- **Support services**

Athabasca University reserves the right to amend course outlines occasionally and without notice. Courses offered by other delivery methods may vary from their individualized study counterparts.

Opened in Revision 7, May 22, 2014

Updated August 31, 2023

View **previous revision**