DATA STRUCTURES

TERM: 2024-I

INSTRUCTORS:

Edwin Andres Niño Velasquez	**e-mail:** eaninov@unal.edu.co	**Group:** 1
David Alberto Herrera Álvarez	**e-mail:** dherreraal@unal.edu.co	**Groups:** 2 and 5
Jhon Alexander López Fajardo	**e-mail:** jalopezfa@unal.edu.co	**Groups:** 3 and 4

MAIN GOAL

To study concepts, techniques, and basic methods on fundamental data structures, as well as their implementation and efficient applications in problem-solving.

SPECIFIC GOALS

- To understand the main data structures (as data collections) and their defined set of operations.
- To implement the main data structures and the operations that can be performed on them in several ways and at least one programming language.
- To solve information management problems through suitable abstraction, detailed design, and optimized implementation to a particular situation.
- To produce implementations of data structures and their operations, contemplating quality aspects such as correction, encapsulation, modularity, efficiency, and simplicity.
- To use other thinking paradigms to design algorithms that allow the students to improve their abilities to solve problems.
- To perform comparative analysis of the main data structures and associated algorithms.
- To be able to understand, apply and code the data structures learned in the solution of computational problems, using the students' own implementations and those standard libraries of the selected programming language.

METHODOLOGY

- This course will combine traditional lecturing with hands-on exercises to reinforce student learning. Students are expected to attend classes regularly, take tests, and submit assignments and other work at the times specified by the instructor.
- A class project will be developed in groups of students to reinforce the application of the main data structures concepts and algorithms. Thus, students will be required to work in teams. Individual class projects will not be allowed, only under exceptional conditions and with a proper justification, such as a medical condition with the corresponding clinical concept. The final deliverable of the term project will be posted in a project fair in an on-line platform.

- Students are encouraged to get engagement for the success of the project and the class. Also, when experiencing inconveniences, conflicts or tension, and having trouble handling such situations please inform this to your instructor as soon as possible.

CONTENTS

- **Introduction**
  - Syllabus Presentation
  - Object Orientation Overview and Introduction to Programming Judges
  - Introduction to Data Structures.
  - Asymptotic Analysis
- **Sequential data structures**
  - Linked Lists
  - Queues, Stacks and Circular Arrays
  - Dynamic Arrays and Amortized Analysis
- **Trees**
  - Tree Terminology
  - Tree Traversals (Inorder, Preorder, Postorder, Level Order)
  - Binary Search Trees (BST)
  - AVL Trees
  - Heaps
  - Heap-Sort
  - Disjoint Sets
- **Hashing**
  - Hashing and Hash Functions
  - Hash Tables
  - Rabin-Karp's algorithm
- **Graphs**
  - Terminology
  - Representation of graphs
  - Operations on Graphs
  - Graph Traversals (Depth First Search- DFS, Breadth First Search-BFS, A*)

STUDENT CONDUCT CODE

Everybody in the class is expected to show courtesy, civility, and respect for one another. Students are expected to do their own work. Cheating, plagiarism, and any other form of academic dishonesty will not be tolerated. Students are encouraged to discuss ideas and techniques broadly with other class members, but not specifics of assigned problems. Sharing of code or intermediate designs for the assigned problems is expressly prohibited. This will be a zero-tolerance policy.

In the assignments to be developed in groups, not doing your fair share of work, or allowing other students to receive credit for an assignment in which they did not work on is considered an act of academic dishonesty.

It is permissible to use software and materials available from other sources (understanding that you get no credit for using the work of others on those parts of your problems) as long as: 1) You acknowledge explicitly which aspects of your assignment were taken from other sources and what those sources are, and 2) The materials are freely and legally available. 3) The material was not created by a student of this course this year or in prior years.

RULES FOR THE USE OF ARTIFICIAL INTELLIGENCE.

The student is allowed to use Artificial Intelligence (AI) tools, such as ChatGPT, to help gather information and generate ideas or source code but may not copy and paste information directly from the AI tool and present it as their own without citing it. The student is responsible for the information he or she submits based on an AI query and for ensuring that it does not contain misinformation, unethical content, or violate intellectual property laws. Your use of artificial intelligence tools must be properly documented and cited to ensure academic integrity. You are expected to include a statement in your assignment describing which AI tool you used and how you used it. For example, "ChatGPT was used to draft approximately 50 percent of this document and to provide review assistance. AI-produced content was edited for accuracy and style".

COMMUNICATION CHANNELS

The following channels will be used to facilitate the communication among the class participants. Also, all the participants will be asked to respect the purpose and rules of each one of these channels.

- **Moodle,** Learning Management System (LMS): This platform will be used as the general channel to post and submit the assignments. In addition, it will be used to administer tests and workshops.
- **UNCode**: This is an on-line judge which will be used to post most of the programming assignments. For such assignments this will be the means to submit the programming deliverables.
- **HackerRank, Leetcode**: These are alternative on-line judges which will be used to post some of the programming assignments.
- **Github**: It will be used as the suggested file repository for your term project.
- **Slack**: Slack will be used as a communication tool. This will be a direct channel to give support and to receive feedback from the students. Students are encouraged to post questions and to help answer your classmates' inquiries via Slack.
- **Whatsapp**: Some instructors may decide to use a WhatsApp group as a messaging tool and as an easier way to communicate. In this case, students are encouraged to post questions and to help answer your classmates' inquiries using this channel.

CLASS ATTENDANCE AND ABSENCES

This course assumes an interactive approach in its structure and in its presentation, which requires engaged participation from all members of the class. Your presence is essential to the success of this course and consequently to your individual success in it. Therefore, regular attendance is expected and considered mandatory.

Please be aware of some penalties that may result from missing classes, such as in-class activities, take-home assignments, or project deadlines.

GRADING POLICY

The evaluation system consists of a point system, in which students will need to accumulate at least 300 points to get a passing grade of 3.0 (out of 5.0). If someone achieves 500 points or more, their corresponding grade will be 5.0. Since the number of quizzes or homework is not bounded, it is possible to accumulate more than 500 points. Each instructor expects to have quizzes, review quizzes or exercises, particular lab exercises and other tasks to complete a minimum of 20 points.

Important: In the practical exercises (programming labs), given that for one assignment different implementations are possible, only credit will be granted to the solutions that meet the requirements for the use of data structures established in the exercises. Also, an object oriented approach is expected to be followed in the solutions of all the programming assignments.

Grading

Assessment		Points	Percentage
Exam I		75	**15%**
Exam II		75	**15%**
Exam III		50	**10%**
Lab Assignments	**Lab:** programming strategies	10	**2%**
	Lab: Object orientation concepts Review Lab.	15	**3%**
	Individual project proposal	20	**4%**
	Lab: implementation of some linear data structures-based containers for the project first prototype	30	**6%**
	First project report: problem definition, goals, and submission of first prototype including the application of basic functions using Linear Data Structures (arrays, lists, queues and stacks)	50	**10%**
	Lab: implementation of trees	35	**7%**
	Second project report: submission of second prototype including the application of trees and graphs.	60	**12%**
	Final Project Report: Submission of final prototype application of all data structures, including hashing based)	60	**12%**
Quizzes and Classwork		>20	**>4%**

Total	>500	>100%

* **Important note:** to meet the objectives of the course, the credit for partial deliveries on the project development will only be taken into account when the final project report is delivered. To make it clear, if the final delivery of the project does not occur, the total grade assigned to this course activity will be zero.

BIBLIOGRAPHY

- M. A. Weiss, Data Structures and Algorithm Analysis in Java, 3rd Ed., Pearson, 2012.
- L. Joyanes and I. Zahonero. Estructuras de datos en Java, 1st Ed., McGraw-Hill, 2008.
- C. A. Shaffer, Data Structures and Algorithm Analysis, Edition 3.2 (Java Version), Virginia Tech, 2012.
- H. M. Deitel and P. J. Deitel, Java How to Program, 9th Ed., Prentice Hall, 2012.
- R. Sedgewick and K. Wayne, Algorithms, 4th Edition, Princeton University, Addison-Wesley, 2011.
- S. Sahni, Data Structures, Algorithms and Applications in Java, Silicon Press; 2nd edition, 2004.
- Aho, J. Hopcroft and J. Ullman, Data Structures and Algorithms,Addison-Wesley, 1983.
- Streib, J. T., & Soma, T. (2017). Guide to Data Structures: A Concise Introduction Using Java. Springer.

OTHER RESOURCES:

- Udemy - Java Tutorial for Complete Beginners: https://www.udemy.com/java-tutorial/learn/#/
- Open Data Structures (in pseudocode) Edition 0.1Gβ Pat Morin: http://opendatastructures.org/ods-python-screen.pdf
- Data Structures and Algorithms in C++. Michael T. Goodrich, Roberto Tamassia, David M. Mount. http://cpp.datastructures.net/
- MIT Introduction to Algorithms lecture videos. Srini Devadas. http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-fall-2011/lecture-videos/
- Pinzón Yoan. Estructuras de Datos. Notas de clase. 2001. Discponible en: https://drive.google.com/file/d/0B43mEy9bxrsQUkpuWERFcXNTUXc
- VisuAlgo - visualising data structures and algorithms through animation. https://visualgo.net/
- Java Downloads
- Code Conventions for the Java Programming Language
- The Java Tutorials
- Java Video Tutorials
- Eclipse - Eclipse Tutorial - Eclipse and Java for Total Beginners (Video Tutorial)
- NetBeans
- PDF creation Software: CutePDF - PDF Creator
- Sumatra PDF
- Java quick reference: 1 sheet - with packages information
- Citing your sources (PDF - Universidad de Los Andes.
- HackerRank.
- Codecademy: Learn Java.
- Coursera: Data Structures. University of California San Diego. National Research University Higher School of Economics.

- CS 61B Data Structures. University of California, Berkeley. Spring 2018.
- Introduction to Algorithms (SMA 5503). Massachusetts Institute of Technology MIT. Electrical Engineering and Computer Science. Fall 2005.
- CS3110 Spring 11 : Data Structures and Functional Programming. Cornell University. Spring 2018.
- 02-713 Algorithms & Data Structures (for Scientists). Carnegie Mellon University. School of Computer Science.

Other resources will be posted in the course on Moodle at https://micampus.unal.edu.co/