

MECHTRON 2MD3

Data Structures and Algorithms for Mechatronics

Winter 2022

# 01 Introduction to the Course

Department of Computing and Software

Instructor:

Omid Isfahanialamdari

January 10, 2022

# My Profile

- Omid Isfahaniaamdari, PhD
  - Post-doc researcher @ University of Pisa
  - Research interest: Big Data Processing (especially mobility data)
  - From September 1, 2021, I moved to Canada
  - I am now a seasonal faculty member @ McMaster University
- Contact
  - Office: N/A
  - Email: [isfahano@mcmaster.ca](mailto:isfahano@mcmaster.ca)
  - Office hours: Wednesdays 15:00 - 16:00
  - Please use McMaster e-mail or Teams for all correspondences, NOT Avenue.
- Lectures: Mondays, Wednesdays, Thursdays: 13:30 -14:20

# Teaching Assistants

- Grad TA: Hina Mahmood, [mahmoh9@mcmaster.ca](mailto:mahmoh9@mcmaster.ca)
- Undergrad TA 1: Junbo Huang, [huanj28@mcmaster.ca](mailto:huanj28@mcmaster.ca)
- Undergrad TA2: Yuxing Fang, [fangy32@mcmaster.ca](mailto:fangy32@mcmaster.ca)
  
- Tutorial hours1: Tue 03:30 PM - 04:20 PM
- Tutorial hours2: Fri 03:30 PM - 04:20 PM
  
- You will be given each week's tutorial exercises (and hopefully lecture slides) during the previous week.

# Why this course?

- Course Description: “Advanced programming with emphasis on embedded systems. Program specifications: Pre- and post-conditions, loop and datatype invariants; use of tools to demonstrate correctness. Selecting data structures for implementation of mathematical abstractions. Finite state machines, automata and languages; lexing and parsing. Algorithm analysis (time and space). Modelling of graphs, relations, corresponding algorithms.”
- From your previous courses, you are familiar with programming in C
- In this course:
  - How to design better software?
  - How to efficiently model, store, access and manipulate data?
  - How to design efficient algorithms to solve problems?
    - Why efficient?
  - How to compare algorithms?

# Why this course?

- Relationship with Mechatronics?
  - Embedded systems limitations
    - Size, computational resources, weight, power supplies
- Data structures are ways to:
  - Organize and store data (storage)
  - Access and manipulate the stored data (access)
- In computer science an algorithm is used to describe a finite, deterministic, and effective problem solving method suitable for implementation as a computer program.
- Different data structures lead to different ways to solve a given problem (algorithms)
- Different algorithms may give different efficiency (space and time)

# Learning Objectives

- Students should know and understand:
  - advanced programming constructs, software design, modularity and abstraction
  - elementary data structures (stacks, queues, bit fields, priority queues, search trees, heaps, hash tables, maps and graph representations)
  - analysis of algorithms, proving correctness of programs and program specifications
  - real-world applications of data-structures
  - basic searching algorithms (binary search, search trees, hashing)
  - basic sorting algorithms (bubble sort, merge sort, quick sort)
  - Finite State Automata and Regular expressions

# Learning Objectives

- Students should be able to:
  - analyze the time and space complexity of algorithms
  - apply time/space trade-offs in designing data structures and algorithms
  - given a programming problem or assignment, select suitable data structures from the ones taught during the course, and provide justification for that selection
  - understand implementation issues for the algorithms studied

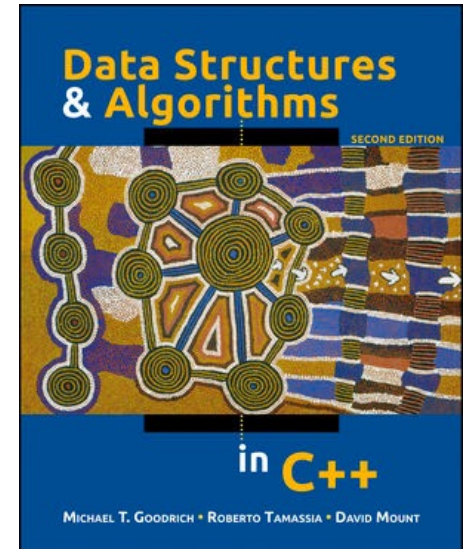
# Tentative List of Topics

- Introduction
- C++ fundamental programming constructs
- Software design concepts, Object-Oriented (OO) design, and abstraction mechanisms
- Fundamental data structures (Stack, Queue, List, Tree, Graph)
- Recursion
- Basic algorithmic analysis
- Proving program correctness
- Searching
- Trees
- Graphs
- Basic sorting algorithms
- Finite-state machines (FSMs), automata and languages



# Resources

- Textbook:
  - Goodrich, Michael T., Roberto Tamassia, and David M. Mount. **Data structures and algorithms in C++**. John Wiley & Sons, 2011.
  - Freely available using your McMaster credentials
- Significant study and reading outside of class is required
- Significant programming exercise is required to grasp the required knowledge
  - Install an IDE of your choice and a C++ compiler and start exercising the codes shown in the lectures



# Marking Scheme

Assignments	30%
Mid-term 1 (date: Feb 16, 2022)	15%
Mid-term 2 (date: Mar 23, 2022)	15%
Final exam	40%

- Assignments
  - Posted on the course web page and submitted through Avenue to Learn.
  - First assignment will be posted on January 21, 2022 (deadline Feb 1, 2022)
  - Grading of assignments will be based on correct fulfilment of all requirements specified in the assignments.
  - Late submissions will be marked with a penalty of 20% per day.
  - Remarking requests will be entertained only within 7 days of the date of return of a graded assignment.
- Final examination: The final examination will be held during the examination period at the end of semester and will cover the entire course contents.

# Academic Integrity

- Students are expected to exhibit honesty and use ethical behavior in all aspects of the learning process.
- Academic dishonesty consists of misrepresentation by deception or by other fraudulent means.
- Academic dishonesty includes:
  - Plagiarism.
  - Copying.
  - Improper collaboration.
- Academic dishonesty can result in serious consequences.
- Your work must be your own. **Plagiarism and copying will not be tolerated!**
- Students may be asked to defend their written work orally.

# Updates on Mid-term and Assignment dates

- **During today:**

- I will make some updates on the count of mid-terms, date of mid-terms and exam dates.
- I will announce the the data of your first assignment.

I solved these issues!  
Please see the relevant slides

# Questions?