Engineering 4892: Data structures and algorithms

Instructor  Jonathan Anderson  TA(s)  Geetha Varsha Chandrasekar
jonathan.anderson@mun.ca  gkosanamchan@mun.ca
(709) 864-2715
Office n/a

Office Hours  Mon 10–11 or by appointment
Website  https://www.engr.mun.ca/~anderson/teaching/4892
Communication  E-mail, Slack and Zoom are preferred. D2L (https://online.mun.ca/d2l/home/352667) will only be used to record grades.

Calendar entry  4892 Data Structures examines fundamental data structures; recursive structures and generic programming techniques; modularity and reusability; time complexity and efficient data structures; procedural abstraction; data abstraction and precise documentation of data structures.

Prerequisite(s)  ENGI 3891
Credit value  3 credit-hours
Accreditation units  4h/week contact hours
Other req.  Tutorial one hour per week

Schedule  Lecture  MWF 15:00–15:50  Zoom
          Tutorial  F 11:00–11:50  Zoom

Textbook  –

Graduate Attribute Focus:  KB (A knowledge base for Engineering), Inv (Investigation) and Des. (Design).

1 Learning Outcomes

Having completed introductory and foundational programming courses and gained some practical experience with C++, you will now explore the world of abstract data types, polymorphic data structures and the algorithmic reasons for using them. After successfully completing this course, you will be able to:

<table>
<thead>
<tr>
<th>Outcome</th>
<th>Attr.Prof</th>
<th>Assessment</th>
</tr>
</thead>
<tbody>
<tr>
<td>develop abstract interfaces to data representations,</td>
<td>PA-D, D-D</td>
<td>assignments, exams</td>
</tr>
<tr>
<td>identify and distinguish the usage of various data structures,</td>
<td>KB-D, PA-D</td>
<td>assignments, exams</td>
</tr>
<tr>
<td>design and implement algorithms using standard data structures,</td>
<td>PA-D, D-D</td>
<td>assignments, exams</td>
</tr>
<tr>
<td>design and implement appropriate data structures for solving computing problems and tasks using appropriate tools,</td>
<td>D-D, T-D</td>
<td>assignments, exams</td>
</tr>
<tr>
<td>design and implement data structures using parametric polymorphism,</td>
<td>D-D</td>
<td>assignments, exams</td>
</tr>
<tr>
<td>test implementations of data structures.</td>
<td>PA-D, I-D, T-D</td>
<td>assignments, exams</td>
</tr>
</tbody>
</table>
2 Assessment

In addition to the typical assignments, mid-term test and final exam, this course’s assessment will include aspects of participation and peer review. We will explore some of this term’s material together in a flipped classroom environment: you will prepare for our time together by reading material, watching video(s) and using interactive tools to explore key concepts. Then, we will spend most of our online time working through exercises, which may be completed collaboratively and will be credited based on timely completion rather than correctness. Grades will be assigned as follows:

- Assignments (5) 25% Individual exploration of key concepts
- Exercises 15% Pro-rated based on completion, not correctness
- Peer code review 15%
- Quizzes (4) 20%
- Final exam 25%

Although exercises may be completed collaboratively, assignments must be completed individually and be a reflection of your own intellectual effort. We will discuss this in class; please don’t hesitate to ask for any clarification that you may need on this policy.

3 Major Topics

1. Review
   - Arrays and vectors
   - Iterators
   - Memory ownership
   - Recursion

2. Algorithmic complexity
   - Asymptotic complexity
   - Big-Θ, Big-Ω, Big-Θ
   - Searching and sorting

3. Abstract data types
   - Generic programming in C++ (templates)
   - Standard template library

4. Linked lists
   - Stacks
   - Queues
   - Skip lists

5. Algorithmic problem solving
   - Greedy algorithms
   - Divide-and-conquer algorithms
   - Dynamic programming
   - Randomized algorithms

6. Sorting algorithms
   - Bubble sort
   - Insertion sort
   - Mergesort
   - Quicksort
   - Bucket sort

7. Hash tables/maps
   - Hash functions
   - Collision resolution
   - Sets and maps

8. Trees
   - Binary search trees
   - Tree traversal
   - Heaps / priority queues
   - optional: AVL, Quadtree, Red-Black, Splay, Tries

9. Graphs
   - Shortest path (Dijkstra, Floyd–Warshall)
   - Minimum spanning tree (Kruskal, Prim)
   - Centrality
   - Dominating set

10. Complexity classes
4 Academic Integrity and Professional Conduct

Students are expected to conduct themselves in all aspects of the course at the highest level of academic integrity. Any student found to commit academic misconduct will be dealt with according to the Faculty and University practices. More information is available at [http://www.mun.ca/engineering/undergrad/academicintegrity.php](http://www.mun.ca/engineering/undergrad/academicintegrity.php).


5 Inclusion and Equity

Students who require accommodations are encouraged to contact the Glenn Roy Blundon Centre, [http://www.mun.ca/blundon/about/index.php](http://www.mun.ca/blundon/about/index.php). The mission of the Blundon Centre is to provide and co-ordinate programs and services that enable students with disabilities to maximize their educational potential and to increase awareness of inclusive values among all members of the university community.

The university experience is enriched by the diversity of viewpoints, values, and backgrounds that each class participant possesses. In order for this course to encourage as much insightful and comprehensive discussion among class participants as possible, there is an expectation that dialogue will be collegial and respectful across disciplinary, cultural, and personal boundaries.

6 Student Assistance

Student Affairs and Services offers help and support in a variety of areas, both academic and personal. More information can be found at [http://www.mun.ca/student](http://www.mun.ca/student).