EXPANDED COURSE DESCRIPTION
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
Lassonde School of Engineering
Electrical Engineering Computer Science
LE / EECS 2011 3.0 SECTION M
FUNDAMENTALS OF DATA STRUCTURES
FALL 2018 / WINTER 2019

Last Modified Date: 08/20/2018

COURSE CALENDAR DESCRIPTION

A study of fundamental data structures and their use in the efficient implementation of algorithms. Topics include abstract data types, lists, stacks, queues, trees and graphs. Prerequisites: cumulative GPA of 4.50 or better over all major EECS courses (without second digit "5"); LE/EECS 1030 3.00 or LE/EECS 2030 3.00; LE/EECS 1028 3.00 OR SC/MATH 1028 3.00 or LE/EECS 1019 3.00 or SC/MATH 1019 3.00. Previously offered as: LE/CSE 2011 3.00.

INSTRUCTOR(S)

<table>
<thead>
<tr>
<th>Name</th>
<th>Section / Format / Term</th>
<th>Contact Email</th>
<th>Contact Phone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mirzaian, Andranik</td>
<td>Sec. M / LECT / W</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ADDITIONAL INFORMATION

Description:
This course introduces fundamental data structures underlying widely-used algorithms. They include various implementations of arrays, lists, maps, hash tables, priority queues, search trees, graphs, and their algorithmic applications. We express these structures in an Object-Oriented Programming (OOP) context and use the Java Programming Language for this purpose.

The course discusses a number of key concepts in OOP. Abstraction and encapsulation are two such concepts. Abstraction at the data level gives rise to expressing a data structure as an Abstract Data Type (ADT). The concept of data abstraction (ADT) predates OOP. OOP adopts a higher level of abstraction, namely objects. A data structure, like any other abstracted entity, is encapsulated as an object with state (data) and behavior (functionality). An object is an instance of its class type. This facilitates a higher level of procedural abstraction: hierarchical inheritance & polymorphism. Parameters in method calls can now be objects of any specified type, possessing not only pure data, but also specified functional behavior.

An ADT's client is any (user application) class that accesses and invokes the ADT. Through encapsulation, a client can directly access only externally visible members of the ADT, namely its Application Programming Interface (API), and is oblivious to the internal details, hence to any particular implementation, of the ADT. The interaction between the client and the ADT implementation is through this API which acts as a contract between them. This contract is expressed by public & protected class member signatures, pre/post conditions, and invariants. Implementing an API means implementing the corresponding ADT that respects the API contract. An important benefit is code flexibility: the ADT implementation can be changed and improved over time without changing its API, and hence, without breaking any client code.

The relationship between this course and your previous courses is as follows:
• EECS 1020: students are clients who use a given API (reading API specs & creating client programs that use them).
• EECS 1030: students are asked to implement a given API.
• EECS 2011: students are asked to design & build a correct & efficient implementation of an ADT to be used by its clientele.
Outcomes:
By the end of the course, students will be familiar with the more prevalent data structure patterns, and will be able to design and implement variations on these patterns, and then use them as clients to solve a broad range of real-world problems.

The weight distribution of the course components is as follows:
• 30% - 4 Assignments, all equal weight
• 25% - Midterm Test - closed book
• 45% - Final Exam - closed book

You will require the following textbook:

LAND ACKNOWLEDGEMENT
We acknowledge our presence on the traditional territory of many Indigenous Nations. The area known as Tkaronto has been care taken by the Anishinabek Nation, the Haudenosaunee Confederacy, the Huron-Wendat, and the Métis. It is now home to many Indigenous Peoples. We acknowledge the current treaty holders, the Mississaugas of the New Credit First Nation. This territory is subject of the Dish With One Spoon Wampum Belt Covenant, an agreement to peaceably share and care for the Great Lakes region.
• The Indigenous Framework for York University: A Guide to Action can be found here: http://indigenous.info.yorku.ca/
• Meaning of a land acknowledgement: http://healthydebate.ca/opinions/indigenous-land-acknowledgements

ACADEMIC INTEGRITY LINKS
• Senate Policy on Academic Honesty - http://secretariat-policies.info.yorku.ca/policies/academic-honesty-senate-policy-on/
• Academic Integrity - http://lassonde.yorku.ca/academic-integrity

STUDENT LINKS
• Student Rights and Responsibilities - http://oscr.students.uit.yorku.ca/student-conduct
• Religious Observance - https://w2prod.sis.yorku.ca/Apps/WebObjects/cdm.woa/wa/regobs
• Academic Accommodation for Students with Disabilities - http://secretariat-policies.info.yorku.ca/policies/academic-accommodation-for-students-with-disabilities-policy/
• Counselling and Disability Services - http://cds.info.yorku.ca/
• York University’s Policies on Sexual Violence - http://secretariat-policies.info.yorku.ca/policies/sexual-violence-policy-on/
• York University’s Policies on Gender/LGBTQ*/Positive Space - http://rights.info.yorku.ca/lgbtq/

Many courses utilize Moodle, York University's course website system. If your course is using Moodle, click here to access it.
Moodle @ York University