CS 234: Data Types and Structures

University of Waterloo

Term and Year of Offering: Spring 2017

Course Number and Title: CS 234, Data Types and Structures

Lecture Times, Building and Room Number:

  Tuesdays and Thursdays, 8:30-9:50am MC2017 and 2:30-3:50pm MC 4020

Instructor’s Name, Office Location, Contact Information, Office Hours:

  Reid Charles Kerr, DC 2133, rckerr@uwaterloo.ca, Tu. 1:30pm - 2:20pm; Th. 10:00am - 11:00am

IA’s Names, Office Location, Contact Information, Office Hours:

  John Wittnebel, DC 3127, john.wittnebel@uwaterloo.ca, Friday 10:00am – 12:00pm
  Ke Nian, DC 3594, knian@uwaterloo.ca, Wednesday 2:00pm – 3:00pm

Course Description:

Top-down design of data structures. Using representation-independent data types. Introduction to commonly used data types, including lists, sets, mappings, and trees. Selection of data representation.

Course Objectives: At the end of the course you should be able to:

  Explain the benefits of using abstract data types (ADTs) and data structures, and more generally, the concepts of modularity and data hiding, in solving problems in a wide variety of application areas. Understand the differing points of view of the user and provider of an ADT and the use of an interface to communicate between views.

  Explain and use the concepts of asymptotic running time, order notation, worst-case, average-case, and best-case complexity, and lower and upper bounds. Given a simple algorithm written in pseudocode, determine and informally justify its asymptotic running time, as expressed in order notation.

  Explain the basic structure of memory, including the distinction between external and internal memory and the concept of paging.

  Explain the advantages and disadvantages of data structures using contiguous or linked memory.

  Explain common ADTs (e.g., Set, Stack, Queue, List, Dictionary, Priority Queue, Tree, Graph), and give a sample application for each. Explain the complexity of some common implementations of these ADTs.

  Explain the distinction between different types of data (e.g. arbitrary, orderable, distinct, digital) and the types of ADTs that can be used for each situation. Given a well-specified computational problem, determine which (if any) ADTs would be suitable.

  Explain how specialized data structures can lead to efficient solutions to specific real-world problems.

  Write, test, and debug Python programs using the material described above.
Optional Text: Rance D. Necaise  Data Structures and Algorithms Using Python. This book is on reserve at the Davis Centre Library.

Readings: Supplementary reading material will be available on the course website.

**Topics to be covered in lectures:**

**Module 1: Introduction**
Topics: Introduction, multiple viewpoints, basic definitions, course goals.

**Module 2: ADTs, pseudocode, and algorithm analysis**
Topics: Recipes for choosing and designing ADTs, types of ADTs and data, pseudocode, different types of analysis, asymptotic notation.

**Module 3: Data structures, ADTs with no order of any kind, Python review**
Topics: Memory, arrays and linked lists, ADT Multiset, Python review, ADT Set, special case of orderable data, sorting.

**Module 4: ADTs with order imposed by operations**
Topics: ADT Stack, ADT Queue, ADT List, ADT Ranking, ADT Superlist, ADT Grid.

**Module 5: ADTs with items related by structure**
Topics: Tree terminology, ADT Unordered Tree, ADT Binary Tree, ADT Ordered Tree.

**Module 6: ADT Dictionary**
Topics: Binary search, tree implementations, external searching, B-trees.

**Module 7: Digital data and average case analysis**
Topics: Decision tree lower bounds, hashing, digital sorting, interpolation search, Move to Front, Transpose.

**Module 8: ADT Priority Queue**
Topics: Array and linked implementations, heaps, heapsort.

**Module 9: ADT Graph**
Topics: Graph terminology, data structures, traversals.

**Module 10: Wrap-up and advanced topics**
Topics: Use of ADTs, ADTs for special situations.
Marks in the course will be calculated as follows:

Assignments: 30%

Midterm: 25%

Final exam: 45%

Students should periodically check recorded marks (using MarkUs) for accuracy.

Notes:
You must pass the weighted exam average in order to pass the course.

Each assignment has the same weight towards the overall mark.

Assignment policies:
All assignments must be completed individually in this course. The solutions you submit must be entirely your own work. Do not look up full or partial solutions on the Internet or in printed sources.

No late assignments will be accepted.

Assignments will be returned in class by having students retrieve their own papers from piles of papers sorted in alphabetical order by last name.

Assignments consist of written and programming components.

Submit both types of components electronically using MarkUs.

Academic Integrity, Grievance, Discipline, Appeals and Note for Students with Disabilities: see www.uwaterloo.ca/accountability/documents/courseoutlinestmts.pdf The text for this web site is listed below:

Academic Integrity: In order to maintain a culture of academic integrity, members of the University of Waterloo community are expected to promote honesty, trust, fairness, respect and responsibility. [Check www.uwaterloo.ca/academicintegrity/ for more information.]

Grievance: A student who believes that a decision affecting some aspect of his/her university life has been unfair or unreasonable may have grounds for initiating a grievance. Read Policy 70, Student Petitions and Grievances, Section 4, www.adm.uwaterloo.ca/infosec/Policies/policy70.htm. When in doubt please be certain to contact the department’s administrative assistant who will provide further assistance.
Discipline: A student is expected to know what constitutes academic integrity [check www.uwaterloo.ca/academicintegrity/] to avoid committing an academic offence, and to take responsibility for his/her actions. A student who is unsure whether an action constitutes an offence, or who needs help in learning how to avoid offences (e.g., plagiarism, cheating) or about “rules” for group work/collaboration should seek guidance from the course instructor, academic advisor, or the undergraduate Associate Dean. For information on categories of offences and types of penalties, students should refer to Policy 71, Student Discipline, www.adm.uwaterloo.ca/infosec/Policies/policy71.htm. For typical penalties check Guidelines for the Assessment of Penalties, www.adm.uwaterloo.ca/infosec/guidelines/penaltyguidelines.htm.

Appeals: A decision made or penalty imposed under Policy 70 (Student Petitions and Grievances) (other than a petition) or Policy 71 (Student Discipline) may be appealed if there is a ground. A student who believes he/she has a ground for an appeal should refer to Policy 72 (Student Appeals) www.adm.uwaterloo.ca/infosec/Policies/policy72.htm.

Note for Students with Disabilities: AccessAbility Services, located in Needles Hall, Room 1132, collaborates with all academic departments to arrange appropriate accommodations for students with disabilities without compromising the academic integrity of the curriculum. If you require academic accommodations to lessen the impact of your disability, please register with the office at the beginning of each academic term.

Turnitin.com: Text matching software (Turnitin®) will be used to screen assignments in this course. This is being done to verify that use of all materials and sources in assignments is documented. Students will be given an option if they do not want to have their assignment screened by Turnitin®. In the first week of the term, details will be provided about arrangements and alternatives for the use of Turnitin® in this course.

Note: students must be given a reasonable option if they do not want to have their assignment screened by Turnitin. See:

https://uwaterloo.ca/academic-integrity/guidelines-instructors for more information.