UNIVERSITY OF CALGARY
FACULTY OF SCIENCE
DEPARTMENT OF COMPUTER SCIENCE
COURSE OUTLINE

1. **Course:** CPSC 331, Data Structures, Algorithms, and Their Analysis -- Fall 2017

Lecture 01: (TR, 09:30-10:45 in ICT116)

Instructor Name	Email	Phone	Office	Hours
Zongpeng Li	zongpeng@ucalgary.ca	403-210-9544	ICT 745	TR 11:00-1:00

Lecture 02: (TR, 14:00-15:15 in ICT114)

Zongpeng Li	zongpeng@ucalgary.ca	403-210-9544	ICT 745	TR 11:00-1:00

Course Site:

D2L: CPSC 331 L01-(Fall 2017)-Data Structures, Algorithms, and Their Analysis

Department of Computer Science: ICT 602, 403 220-6015, cpsc@cpsc.ucalgary.ca

2. **Prerequisites:**

See section 3.5.C in the Faculty of Science section of the online Calendar.

> One of Computer Science 219, 233, 235 or Computer Engineering 339 and one of Mathematics 271 or 273. Credit for Computer Science 331 and 319 will not be allowed.

3. **Grading:**

The University policy on grading and related matters is described in F.1 and F.2 of the online University Calendar. In determining the overall grade in the course the following weights will be used:

Component(s)	Weighting %
Assignments	25%
Midterm (in-Class Tuesday Ocotber 24th, 2017)	30%
Final Exam	45%

Each piece of work (reports, assignments, quizzes, midterm exam(s) or final examination) submitted by the student will be assigned a percentage score. The student's average percentage score for the various components listed above will be combined with the indicated weights to produce an overall percentage for the course, which will be used to determine the course letter grade.

The conversion between a percentage grade and letter grade is as follows;

Letter Grade	A+	A	A-	B+	B	B-	C+	C	C-	D+	D
Minimum Percent Required	95	90	85	80	75	70	65	60	55	50	45

4. **Missed Components of Term Work:**

The regulations of the Faculty of Science pertaining to this matter are found in the Faculty of Science area of the Calendar in Section 3.6. It is the student's responsibility to familiarize himself/herself with these regulations. See also Section E.3 of the University Calendar

5. **Scheduled out-of-class activities:**

There are no out-of-class activities scheduled for this course.

6. **Course Materials:**

   Optional Textbook(s):

   Cormen, Leiserson, Rivest and Stein, Introduction to Algorithms, MIT Press

7. **Examination Policy:**

   No aids are allowed on tests or examinations

   Students should also read the Calendar, Section G, on Examinations.

8. **Approved Mandatory and Optional Course Supplemental Fees:**

   There are no mandatory or optional course supplemental fees for this course

9. **Writing across the Curriculum Statement:**

   See Section E.2 of the University Calendar.

10. **Human studies statement:**

    Students will not participate as subjects or researchers in human studies.

11. **OTHER IMPORTANT INFORMATION FOR STUDENTS:**

    a. **Misconduct:** Academic misconduct (cheating, plagiarism, or any other form) is a very serious offence that will be dealt with rigorously in all cases. A single offence may lead to disciplinary probation or suspension or expulsion. The Faculty of Science follows a zero tolerance policy regarding dishonesty. Please read the sections of the University Calendar under Section K. Student Misconduct to inform yourself of definitions, processes and penalties.

    b. **Assembly Points:** In case of emergency during class time, be sure to FAMILIARIZE YOURSELF with the information on assembly points.

    c. **Academic Accommodation Policy:** Students needing an Accommodation because of a Disability or medical condition should contact Student Accessibility Services in accordance with the Procedure for Accommodations for Students with Disabilities available at procedure-for-accomodations-for-students-with-disabilities_0.pdf.

       Students needing an Accommodation in relation to their coursework or to fulfil requirements for a graduate degree, based on a Protected Ground other than Disability, should communicate this need, preferably in writing, to the Associate Head of Undergraduate Affairs of the Department of Computer Science, Nathaly Verwaal by email nmverwaa@ucalgary.ca or phone 403-220-8485.

    d. **Safewalk:** Campus Security will escort individuals day or night (www.ucalgary.ca/security/safewalk/) . Call 403-220-5333 for assistance. Use any campus phone, emergency phone or the yellow phones located at most parking lot pay booths.

    e. **Freedom of Information and Privacy:** This course is conducted in accordance with the Freedom of Information and Protection of Privacy Act (FOIPP). As one consequence, students should identify themselves on all written work by placing their name on the front page and their ID number on each subsequent page. For more information, see also www.ucalgary.ca/legalservices/foip.

    f. **Student Union Information:** VP Academic, Phone: 403-220-3911 Email: suvpaca@ucalgary.ca. SU Faculty Rep. Phone: 403-220-3913 Email: sciencerep@su.ucalgary.ca; Student Ombudsman, Email: suvpaca@ucalgary.ca

    g. **Internet and Electronic Device Information:** You can assume that in all classes that you attend, your cell phone should be turned off unless instructed otherwise. Also, communication with other individuals, via laptop computers, Blackberries or other devices connectable to the Internet is not allowed in class time unless specifically permitted by the instructor. If you violate this policy, you may be asked to leave the classroom. Repeated abuse may result in a charge of misconduct.

    h. **Surveys:** At the University of Calgary, feedback through the Universal Student Ratings of Instruction (USRI) survey and the Faculty of Science Teaching Feedback form provides valuable information to help with evaluating instruction, enhancing learning and teaching, and selecting courses. Your responses make a difference - please participate in these Surveys.

    i. **SU Wellness Center:** The Students Union Wellness Centre provides health and wellness support for students including information and counselling on physical health, mental health and nutrition. For more information, see www.ucalgary.ca/wellnesscentre or call 403-210-9355.

Department Approval: Electronically Approved **Date:** 2017-09-05 18:32

Course Outcomes

1. Discover assertions that explain why algorithms are correct, and that can be used as documentation or to make programs self-testing.
2. Develop mathematical expressions for time and storage requirements, for simple algorithms, given pseudocode for these algorithms.
3. Explain - clearly and precisely - why algorithms, discussed in this course, are correct and efficient.
4. Use asymptotic notation to simplify expressions for resource requirements of algorithms, without omitting essential information about these.
5. Understand and correctly identify asymptotic relations between functions that are commonly used to bound resource requirements - including logarithmic functions, polynomial functions with various degrees, and exponential functions with various bases.
6. Describe various classical abstract data types including stacks, queues, dictionaries and graphs as well as the operations that each supports.
7. Describe several data structures that can be used to implement each of the abstract data types that have been studied, and compare and contrast the resource requirements for the resource requirements for each operation when each data structure is used.
8. Use simple English descriptions as well as pseudocode to describe algorithms to implement the operations of an abstract data type when a given data structure is used to implement it.
9. Identify the abstract data types that can be used to solve a variety of computations, along with data structures that can be used to implement these when resource bounds for the cost of operations are also supplied.
10. Use a modern (object-oriented) programming language to implement various abstract data types using specified data structures and use a software library - which includes industry-standard implementations of abstract data types with promised resource bounds for the costs of operations - to write short and simple programs that solve nontrivial computation problems correctly, and within specified resource bounds.