Course: CS261- Data Structures in CS
Credits: 4
Instructor's name: Samina Ehsan
Instructor's email: ehsan@eecs.oregonstate.edu
Class email: oregonstatecs261@gmail.com

OSU catalog course description, including pre-requisites/co-requisites:
Complexity analysis, Approximation methods, Trees and graphs, File processing, Binary search trees, Hashing, Storage management. Lec/rec. **PREREQS: (**(ECE 152 or CS 162) and MTH 231**)**

Course Content:

The course content is described as followed –

Week	Course Topics

#1	• Reading: Chapters 1- 4
	• Assignment # 0 : Introduce Yourself and Set up your IDE
	• Worksheet – "Group Participation Strategy"
	• Worksheets 9 and 10 (Review Content.)
	• Video: C_Basics_Review
	• Video: eclipseProjectFromMakefile
	  o Code: studentStructExample
	• Video: C_Pointers_Review
	• Video: Static_dynamic_structCodeExamples
	  o Code: studentStructExample
	  o Code: dynamicStudentStructExample
	• Video: C_Compiling_Review
	• Programming Assignment # 1 – C Pointers Practice

#2	• Reading: Chapter 5
	• Video: AbstractDataTypes
	• Worksheet 0 : Array_Bag_Stack
	  o Solution Code: arrayBagStack
	• Reading: Chapter 6 pp. 1-10
	• Video: DynamicArrayConcepts
	• Worksheet 15 : DynArr_Amortized
	• Video:DynamicArrayImplementation
	• Worksheet 14 : DynArr
	• Worksheet 16 : DynArr_Stack
	• Reading: Chapter 8 pp. 1-4
	• Worksheet 21 : DynArr_Bag
	  o Solution Code: dynamicArray [locked until after assignment 2 turned in]
	• Programming Assignment # 2 - Amortized Analysis and Dynamic Array Stack Application

#3	• Reading: Chapter 7 pp. 1-2, 6-10 • Video: DynamicArrayDequeIntro • Worksheet 20: Dynamic Array Deque and Queue (Read the Introduction) Solution Code: DynamicArrayDeque • Video: DynamicArrayDequeImplementation • Worksheet 20: Dynamic Array Deque and Queue (Complete the implementation) • Reading: Chapter 6 pp. 10 - 19 • Video: LinkedListIntro • Worksheet 17: LinkedList Stack o Solution Code: Linked List Stack • Reading: Chapter 7 pp. 4-6 • Video: LinkedListQueue • Worksheet 18: LinkedList Queue o Solution Code: Linked List Queue • Video:LinkedListDequeue • Worksheet 19: LinkedList Deque o Solution Code: LinkedList • Reading: Chapter 8 pp. 4-9 • Worksheet 22: Linked List Bag • Programming Assignment # 3 - Circular Linked List
#4	• Video: Iterator ADT • Worksheet 23: Dynamic Array Iterator • Worksheet 24: Linked List Iterator • Code Demo Video: Linked List Iterator o Code: LinkedListIterator (Folder) • Reading: Chapter 9 • Video: Ordered Arrays and Binary Search • Worksheet 26: Ordered Bag using Ordered Array • Video or Handout: Binary Search Argument of Correctness • Reading: Chapter 10 pp. 1-5, 13-19 • Video: Trees Intro • Video: BST 1 • Worksheet 28 • Video: BST 2 • Worksheet 29 • Programming Assignment # 4 - Binary Search Trees
#5	• Reading: Worksheet 31 (do not complete yet) • Video: AVL 1 • Video: AVL 2 • Worksheet AVL : AVL Practice • Video: AVL Implementation – code walkthrough o Code: AVL Tree (Folder) • Worksheet 31 – complete the implementation • Reading: Chapter 11 pp. 1-7 • Video: Heaps I • Worksheet Heaps: Heaps Practice • Video: Heaps II • Worksheet 33: Heaps and Priority Queues

	• Reading: Chapter 11 pp. 7 - 14 • Video: Heap Sort • Worksheet 34: Build Heap and Heap Sort • Programming Assignment # 5: Heap Implementation of a To-Do List
#6	• **Midterm Exam : (Week 1- Week 4)** • Reading: Chapter 12: pp. 3-6 • Video: HashTables Intro • Video: HashTables_OpenAddressing • Video: Maps • Code: DynArryMap (Folder) • Worksheet 36 : Dynamic Array Dictionary • • Worksheet 37: Open Address Hashing • Reading: Chapter 12: pp. 6-15 • Video: HashTables_Chaining • Worksheet 38 : HashTables Using Buckets • Video: Hash-Like Sorting • Programming Assignment # 6: Hash Table Implementation of a Concordance •
#7	• Reading: Chapter 13: Graphs • Video: Graphs Intro • Worksheet 40: Graph Representations • Video: GraphAlgorithms I • Worksheet 41: Depth-First and Breadth-First Search • Reading: Chapter 7 pp. 2-4 • Video: GraphAlgorithmsII DFS/BFS • Video: GraphAlgorithms III Dijkstra • Worksheet 42: Dijkstra's Algorithm • More Practice: bfs.pdf, dfs.pdf, dijkstras.pdf • Programming Assignment # 7: Graphs
#8	• Video: BST Iterator • Worksheet 30: Binary Search Tree Iterator • Reading: Chapter 10 pp. 5-13 • Video: Tree Traversals • Worksheet 32: Tree Sort • Redo Worksheet 32 using BST Iterator • Reading: Chapter 12 pp. 1-3
#Final Week	**Final Exam(Week 5 – Week 8)**

Canvas & Piazza — This course will be delivered via Canvas, your online learning community. Within the course Canvas site you will access the learning materials, tutorials, and syllabus; submit assignments; take quizzes and

exams; find your discussion group; email the instructor. To preview how an online course works, visit the Ecampus Course Demo. For technical assistance, Canvas and otherwise, see http://ecampus.oregonstate.edu/services/technical-help.htm. Piazza will be used as the medium for class discussion where you will interact with your classmates and with me. You can discuss issues; participate in online activities. For Piazza, visit https://piazza.com/oregonstate/summer2015/cs261_400/home.

Course Learning Objectives

At the completion of the course, students will be able to...

1. **Describe** the properties, interfaces, and behaviors of basic abstract data types, such as collection, bag, indexed collection, sorted collection, stack, and queue.
2. **Read** an algorithm or program code segment that contains iterative constructs and **analyze** the asymptotic time complexity of the algorithm or code segment.
3. **State** the asymptotic time complexity of the fundamental operations associated with a variety of data structures, such as vector, linked list, tree, and heap.
4. **Recall** the space utilization of common data structures in terms of the long-term storage needed to maintain the structure, as well as the short-term memory requirements of fundamental operations, such as sorting.
5. **Design** and **implement** general-purpose, reusable data structures that implement one or more abstractions.
6. **Compare** and **contrast** the operation of common data structures (such as linear structures, priority queues, tree structures, hash tables, maps, and graphs) in terms of time complexity, space utilization, and the abstract data types they implement.

Learning Resources:

Online: CS261_ClassNotes_Fall2012.pdf *(*will be available on Canvas*)*
C reference book (C Programming Language by Brian W. Kernighan and Dennis M. Ritchie) (optional)
Or
Any reference book to C programming language

Evaluation of Student Performance:

Scores for worksheets, programming assignments, and exams will be posted on Canvas as they are graded. We **will not use** Canvas grading scheme. If you want to know your grade, use the following weights-

- 30% - Homework Assignments
- 15% - Worksheets and Class Participation
- 25% - Midterm
- 30% - Final

Worksheets and Class Participation (15%)

Each week, you will be given 5 worksheets to complete in a group setting through online discussion. These worksheets are very important to your understanding of the material and often contain additional reading material as well as exercises (problems, coding, etc.).

Worksheets are graded primarily based on participation and effort, rather than correctness. If you have made a reasonable effort to complete a worksheet (evidenced by online discussion on Piazza) you will receive full credit for it. Please complete this group forming (maximum 5 people) and setting the medium of collaboration by week 1 and let the Instructor and Teaching Assistants know**. If the discussion is not visible to the Instructor/ Teaching Assistants, you will not receive any grade for it.** No late submission will be allowed**.**

These worksheets are supposed to enhance the lectures using hands-on learning. In most cases, completion of the worksheet will be the first step of the upcoming assignment. Worksheets are designed to be finished in 1-2 hours.

Completing the worksheets within groups is truly rewarding. Solving the worksheet problems together often helps the students to learn how to work in teams. You may find yourself involved in group work not only in later courses you take in this program but also in the career you chose in future. Please put your best effort to make the team work successful.

Homework Assignments (30%)

There are 7 total assignments to be completed over the course of this class.

• Assignments include writing a computer program and sometimes written answers to questions.
• Assignments are to be turned in **before 23:59** on the date they are due. **NOTE**: You are permitted one late programming assignment to use at any time during the quarter. The late assignment must be submitted no more than 48 hours after the original deadline. This means that if an assignment is due on Oct 1 at 23:59, you may turn it in as late as Oct 3 at 23:59.
• Programs are evaluated on how well they solve the assigned problem (adhering to program specification), as well as the proper formatting and use of comments.
• **Programming assignments must compile on FLIP server**. **You will not receive any grade if your assignment doesn't compile.**
• You **must** turn in your assignments through both the Canvas and TEACH websites. **20% of the grade will be deducted if you do not submit it to both sites.**
• If you have a problem with an assignment grade, you must contact the teaching assistant, who graded your assignment, through EMAIL within **ONE WEEK** of receiving your grade.

Exams (55% Total):

There are 2 total exams for this course.

• The midterm is given in **Week 6** and the Final in **Week 9**. Please check the actual dates provided in the **weekly schedule document**. You will be given a 4 days long time window to take each exam. No extension will be allowed outside those assigned windows
• The midterm is designed to take 130 minutes maximum.
• The final is designed to take 110 minutes maximum.
• **Exams must be submitted only to Canvas**.
• Students can use external offline editors to type their answers. They should create a .pdf file of their text editor to upload their exam to Canvas. If they fail to do that, they must take help from the proctor

to send it to the instructor via email. Students must destroy the .pdf file after the submission. No other websites are allowed. Tablet or Notebook is not allowed in the exam.

• The instructor must be informed once the student has completed the exam.

**REMINDER: This course requires that you take the 2 exams under the supervision of an approved proctor. ProctorU is an allowed option for this course. It is entirely the student's responsibility to secure and schedule a proctor before the exam due date and is very important to submit your proctoring request as early as possible to avoid delays. Please remember that late exams will not be allowed due to not having scheduled a proctor early enough. Registration for proctored exams is available online and there is generally a small fee associated with exam proctoring. For more information please visit: <http://ecampus.oregonstate.edu/services/ proctoring/>. If you need assistance please contact ecampustesting@oregonstate.edu or 541-737-9281.

Course Policies:

Grading Policies:

We will use the following grading structure to calculate the final grade-

100 >=A>= 92.5
92 >=A->= 89.5
89 >=B+>= 86.5
86 >=B>= 82.5
82 >=B->= 79.5
79 >=C+>= 76.5
76 >=C>= 72.5
72 >=C->= 69.5
69 >=D+>= 66.5
66 >=D>= 62.5
62 >=D->= 59.5
59 >=F

**** REMINDER**: A passing grade for core classes in CS is a C or above. A C-, 72 or below, is not a passing grade for CS majors.

Exam Policies — Preparing makeup exams requires a significant effort on the part of the instructor. Consequently, makeup exams will not routinely be given. Makeup exams will be given only for missed exams excused in advance by the instructor. For missed exams that can be anticipated ahead of exam time, advance permission from the instructor to miss the exam will be necessary. Excused absences will not be given for airline reservations, routine illness (colds, flu, stomach aches), or other common ailments. Excused absences will generally not be given after the absence has occurred, except under very unusual circumstances. Regrades of exams will be performed when there is an error and the student requests it. All requests for regrading must be made within 3 class days of the day the exam is returned. After that period of time, grades will be fixed and will not be changed.

(BB 450 Instructor: Kevin Ahern)

Incompletes — In this online program, there will rarely be cases where an incomplete is appropriate.
The instructor will only consider giving an incomplete grade for emergency cases such as a death in the family, major disease, or child birth, while also having completed at least 60% of all coursework. If you have a situation that may prevent you from completing the coursework, let the instructor know as soon as you can.

(CS Instructor: Joseph Joss)

Statement Regarding Students with Disabilities:

Accommodations are collaborative efforts between students, faculty and Disability Access Services (DAS) with accommodations approved through DAS are responsible for contacting the faculty member in charge of the course prior to or during the first week of the term to discuss accommodations. Students who believe they are eligible for accommodations but who have not yet obtained approval through DAS should contact DAS immediately at 541-737-4098.

Expectations for Student Conduct:

Student conduct is governed by the university's policies, as explained in the Office of Student Conduct: information and regulations. In an academic community, students and faculty, and staff each have responsibility for maintaining an appropriate learning environment, whether online or in the classroom. Students, faculty, and staff have the responsibility to treat each other with understanding, dignity and respect. Disruption of teaching, administration, research, and other institutional activities is prohibited by Oregon Administrative Rule 576-015-0015 (1) and (2) and is subject to sanctions under university policies, OSU Office of Student Conduct.

Academic Integrity - Students are expected to comply with all regulations pertaining to academic integrity. At OSU academic integrity is defined as the following: "(a) upholding the standards of the academic discipline of which you are a part, (b) honesty in all academic processes and accomplishments, (c) respect for and appropriate use of the work of others, (d) taking responsibility for your own work, and (e) accountability to protect personal academic work from misuse by others."

Academic Dishonesty - is defined as an act of deception in which a Student seeks to claim credit for the work or effort of another person or uses unauthorized materials or fabricated information in any academic work or research, either through the Student's own efforts or the efforts of another. For further information, visit Avoiding Academic Dishonesty, or contact the office of Student Conduct and Mediation at 541-737-3656.

The following two policies apply here:
OSU policy:
http://oregonstate.edu/studentconduct/http%3A/%252Foregonstate.edu/studentcond uct/faculty/facacdis.php
College of Engineering policy:
http://engineering.oregonstate.edu/undergraduate-policy-manual#honesty

Additionally, programming assignments in this course are considered Take Home Programming Tests. You must do your own work, entirely.

- You MAY discuss the meaning of assignments, general approaches, and strategies with other students in the course.
- You MAY show your code to the TAs or instructor for feedback and help.
- You MAY use the Internet to research how to solve a problem.
- You MUST include a citation in the form of a comment in your source code to indicate the source of any help you received (except the TAs).

- You MUST ALSO include a citation if you collaborated with any other student in any way (both the giver and receiver).
- You MAY NOT share assignment code, pseudocode, or documentation of any kind with any other student in the course.
- You MAY NOT show your assignment code to another student in the course for any reason.
- You MAY NOT ask another student for help debugging your assignment code.
- You MAY NOT use or copy code from any other source, including the Internet.
- You MUST write your own code for your assignments.

(Adapted from statements provided by Dr.Ronald Metoyer, CS)

Conduct in this online classroom — Students are expected to conduct themselves in the course (e.g., on discussion boards, email postings) in compliance with the university's regulations regarding civility. Students will be expected to treat all others with the same respect as they would want afforded themselves. Disrespectful behavior to others (such as harassing behavior, personal insults, inappropriate language) or disruptive behaviors in the course (such as persistent and unreasonable demands for time and attention both in and out of the classroom) is unacceptable and can result in sanctions as defined by Oregon Administrative Rules Division 015 Student Conduct Regulations.

(Adapted from statements provided by Becky Warner, SOC)

Communications:

Ground Rules for Online Communication & Participation:

- *Online threaded discussions* are public messages, and all writings in this area will be viewable by the entire class or assigned group members. If you prefer that only the instructor sees your communication, send it to me by email, and be sure to identify yourself and the class.
- Posting of personal contact information is discouraged (e.g. telephone numbers, address, personal website address).
- *Online Instructor Response Policy:* I will check email frequently and will respond to course-related questions within 24 hours.
- *Observation of "Netiquette":* All your online communications need to be composed with fairness, honesty and tact. Spelling and grammar are very important in an online course. What you put into an online course reflects on your level of professionalism. Here are a couple of references that discuss
  - writing online: http://goto.intwg.com/
  - netiquette: http://www.albion.com/netiquette/corerules.html.
- Please check the Announcements area and the course syllabus before you ask general course "housekeeping" questions (i.e. how do I submit assignment 3?). If you don't see your answer there, then please contact me.

(Adapted from Jean Mandernach, PSY)

Guidelines for a productive and effective online classroom

- Piazza is your space to interact with your colleagues related to current topics or responses to your colleague's statements. It is expected that each student will participate in a mature and respectful fashion.
- Participate actively in the discussions, having completed the readings and thought about the issues.
- Pay close attention to what your classmates write in their online comments. Ask clarifying questions, when appropriate. These questions are meant to probe and shed new light, not to minimize or devalue comments.

- Think through and reread your comments before you post them.
- Assume the best of others in the class and expect the best from them.
- Value the diversity of the class. Recognize and value the experiences, abilities, and knowledge each person brings to class.
- Disagree with ideas, but do not make personal attacks. Do not demean or embarrass others. Do not make sexist, racist, homophobic, or victim-blaming comments at all.
- Be open to be challenged or confronted on your ideas or prejudices.

(Adapted from a statement provided by Susan Shaw, WS)

Student Assistance:

Contacting the instructor —

- Sending email (ehsan@eecs.oregonstate.edu) is the best way to ask any question related to the course. If needed, we can use Skype or Google Hangout for discussion.
- Piazza is another preferred way to reach the instructor and TAs for any course related query. We can refer back to our previous discussions here and also as it will be visible to the entire class. So, the other students will be able to get benefit from it.
- We will have constant TA support, so it should be possible to get help at any time. Use the class account - oregonstatecs261@gmail.com to get support during the office hours. • I will maintain virtual office hours using the same class account.

Technical Assistance — If you experience computer difficulties, need help downloading a browser or plug-in, assistance logging into the course, or if you experience any errors or problems while in your online course, contact the OSU Help Desk for assistance. You can call (541) 737-3474, email osuhelpdesk@oregonstate.edu or visit the OSU Computer Helpdesk online.

Tutoring — Effective fall term 2009 we went to a new Online Tutoring Service - NetTutor to meet the needs of Ecampus students.

> NetTutor is a leading provider of online tutoring and learner support services fully staffed by experienced, trained and monitored tutors. Students connect to live tutors from any computer that has Internet access. NetTutor provides a virtual whiteboard that allows tutors and students to work on problems in a real time environment. They also have an online writing lab where tutors critique and return essays within 24 to 48 hours.

Course Evaluation:

I will encourage that a student will be able to, anonymously, make comments, requests, or suggestions in regards to the design and implementation of the content of the course.

> **OSU Student Evaluation of Teaching** — Course evaluation results are extremely important and are used to help me improve this course and the learning experience of future students. Results from the 19 multiple choice questions are tabulated anonymously and go directly to instructors and department heads. Student comments on the open-ended questions are compiled and confidentially forwarded to each instructor, per OSU procedures. The online Student Evaluation of Teaching form will be available toward the end of each term, and you will be sent instructions by Ecampus. You will login to "Student Online Services" to respond to the online questionnaire. The results on the form are anonymous and are not tabulated until after grades are posted.

Concluding Remark:

> Get your data structures correct first, and the rest of the program will write itself.”
>
> —Davids Johnson

Please take the above quote seriously.