Data Structures Course Syllabus

Subject Code and Course Number:	CS 121, section C
Course Title:	Data Structures
Number of Credits:	3 credits
Instructor Name:	Varduhi Yeghiazaryan
E-mail Address:	vyeghiazaryan@aua.am
Office Location:	331W
Office Hours:	Monday 15:00–16:00, Thursday 15:00–16:00, or by appointment

Teaching Associate:	Tigran Hayrapetyan
TA Problem Solving Session:	Lab003, Wednesday 16:30–18:30
TA E-mail Address:	tigran.human@gmail.com
TA Office:	310W
TA Office Hours:	Monday, 14:30–16:00

Term/Year:	Fall 2019
Class Schedule:	Tuesday, Thursday 12:00–13:15

Prerequisites:	CS 111, CS 120
Co-Requisites:	None

Course Description: The course explores topics including: basic object-oriented programming principles; linear and non-linear data structures – linked lists, stacks, queues, trees, tables and graphs; dynamic memory management; design of algorithms and programs for creating and processing data structures; searching and sorting algorithms. Students are required to complete programming projects in which they design, analyze, and develop complex data structures in at least one programming language. Three hours of instructor-led class time per week including discussions and problem sets.
Subject to Change to accommodate student needs.

Required Materials:

Main textbooks:

Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser. Data Structures and Algorithms in Java, 6th edition, John Wiley and Sons, 2014.

Alternate Version of textbook:

Michael T. Goodrich, Roberto Tamassia, David M. Mount. Data Structures and Algorithms in C++, 2nd edition, John Wiley and Sons, 2011.

Schedule & Topics (subject to possible changes)

*Assignments may be supplemented to address student needs identified through assignments, quizzes and the midterm exams.

* Home tasks will be provided in Moodle in a separate PDF file.

Week	**Topic**	**Reading**	**Non-Reading Home Tasks**
1	Introduction to Data Structures, Asymptotic Analysis	Ch. 2, Ch. 4	HW1 Released
2	Recursion	Ch. 5	
3	Search and Sorting	Ch. 3	HW2 Released
4	Sorting, Linked Lists	Ch. 12, Ch. 3	
5	Linked Lists, Stacks **QUIZ 1**	Ch. 3, Ch. 6	HW3 Released
6	Queues, Vectors/Lists	Ch. 6, Ch. 7	
7	Vectors, Lists, Iterators **MIDTERM 1 (Oct. 12)**	Ch. 7	HW4 Released
8	Vectors, Lists, Iterators, Trees	Ch. 7, Ch. 8	
9	Trees	Ch. 8	HW5 Released
10	Priority Queues, Heaps, Heapsort **QUIZ 2**	Ch. 9	
11	Priority Queues, Heaps, Heapsort, Maps	Ch. 9, Ch. 10	
12	Search Trees, Balanced Trees **MIDTERM 2 (Nov. 16)**	Ch. 8, Ch. 11	HW6 Released
13	Balanced Trees, Hash Tables	Ch. 11	
14	Hash Tables **QUIZ 3**	Ch. 10	
15	REVIEW, Q&A		

Tentative Dates of Quizzes and Midterm Exams

Examinations	Week	Date
QUIZ 1	5	TBA
MIDTERM 1	7	October 12 at 11:50 – 13:50 in LA
QUIZ 2	10	TBA
MIDTERM 2	12	November 16 at 12:30 – 14:30 in LA
QUIZ 3	14	TBA

Student Learning Outcomes:
The following chart shows alignment between course-specific and program student learning outcomes and program goals.

Course-based Student Learning Outcomes *In this course, students will be able to:*	***Program Student Learning Outcomes*** *Students will be able to:*	***Program Goal***
Develop in-depth understanding of the concept of data abstraction	1.3. Utilize and adapt software and select and use hardware systems related to computer science (Beginner Level)	Equip students with knowledge and advanced skills in mathematical reasoning, problem solving, modeling and scientific computation
Build mastery in important data structures such as lists, stacks, queues, priority queues, heaps, hash tables, trees and graphs Adapt and apply the basic principles of object-oriented programming Develop robust and efficient code in at least one programming language	3.1. Design and analyze complex data structures and algorithms 3.2. Develop and implement software applications in one or more programming languages 3.3. Develop and test software tools and methods (Intermediate Level)	Prepare students for development of scientific, engineering and industrial software applications
Identify and use appropriate data structures and sorting and searching algorithms	5.1. Use common software and information technology to pursue inquiry relevant to their academic and professional fields, and personal interests (Advanced Level)	Provide students with a broad foundation of knowledge and skills and cultivate a commitment to life-long learning

Course Structure: Instructor-led class will meet twice a week Tuesday and Thursday at 12:00-13:20. Students are encouraged to attend these classes, as well as a weekly problem solving sessions. Home tasks include reading and programming assignments.

Method of Evaluation: Students progress will be assessed by homework assignments, quizzes, midterm exams and final exam.

Homework: Homework assignments will be given on a weekly or bi-weekly basis except for the last and exam weeks. Homework problems will be assigned by the instructor throughout the semester and will be posted on Moodle. Students are to submit their work **electronically** before the deadline. **No late homework will be accepted.** The format of submitting homework assignments will separately be posted as a PDF on Moodle.

Any programming involved in any of the homework assignments must be coded in either Java or C++.

Homework collaboration policy:

- You are encouraged to study and discuss problems with fellow students, but you must write and code your own homework by yourself, and make sure you understand how to obtain the solution to the problems.
- Any collaboration should be explicitly announced. In such a case, the homework assignment will be graded by 70% of the actual grade obtained on the said homework for all mentioned student. Collaboration is allowed with a group of maximum 3 students. **Unacknowledged collaboration will be graded 0.**

Quizzes: The course will include three quizzes. The quizzes will be closed-book and closed-device: use of calculators, computers, tables, phones are prohibited; books and notes may not be consulted. Collaboration during quizzes is strictly forbidden.

Midterm Exams: The course will include two midterm exams. The midterms are closed-book and closed-device: use of calculators, computers, tables, phones are prohibited; books and notes may not be consulted. Collaboration during exams is strictly forbidden.

Final Exam: The final exam will cover all topics covered throughout the course. The purpose of the final exam is to assess students' mastery of concepts and terminology learned during class as well as their ability to apply this knowledge to practical problems.

For any of the quizzes and exams, students are required to choose only one of the two languages: Java or C++.

Moodle: You should enroll into the course on Moodle. Homework assignments, learning material and students' grades will be posted regularly on Moodle. Thus, please check the course's Moodle page regularly.

Method of Evaluation: Student learning will be evaluated on the basis of the following weighted components:

• Homework:	15%
• Quizzes:	9%
• Midterms:	36%
• Final Exam:	40%

The final grades will be defined according to the following ranges given below.

Grade	Grade Point	Percentile Range
A+	4	95 –≤100
A	4	90 – < 95
A-	3.7	85 – < 90
B+	3.3	80 – < 85
B	3	75 – < 80
B-	2.7	70 – < 75
C+	2.3	66 – < 70
C	2	62 – < 66
C-	1.7	58 – < 62
D+	1.3	55 – < 58
D	1	53 – < 55
D-	0.7	50 – < 53
F	0	0 – < 50

Library and Media/Technology Use

Students are encouraged to use supplemental online and reference materials available at the library to enhance their overall learning in the course. If students have any questions or need additional support in using library resources, they should confer with library staff, ICT, or the instructor.

Late Policy

Students are required to submit homework assignments by the deadline scheduled by the instructor. Late submitted homework will not qualify for grading.

Make-up Procedures

The make-up procedure can be organized only for missed exams at the instructor's discretion. The student must submit convincing evidence of a medical or other emergency that makes taking the exam at the scheduled time impossible. It is also important to notify the instructor in advance on cases when taking exam in time is impossible for whatever reasons.

Policy on Grade Appeal

Students are entitled to appeal grades in line with the university's Grades Policies policy which is available online at http://policies.aua.am

Standards for Academic Integrity

Students are required to conduct themselves in an academically responsible and ethical manner in line with the Student Code of Ethics. Acts of academic dishonesty impair the academic integrity of AUA and create an unfair academic advantage for the student involved and other member(s) of the academic community. These acts are subject to disciplinary measures as prescribed in the AUA Code of Student Ethics http://policies.aua.am/policy/10

The Student Code of Conduct can be found at http://policies.aua.am/policy/101

Special Needs:

Students requiring special accommodations for learning should contact the Center for Student Success by the end of the Drop/Add period with such requests. studentsuccess@aua.am, http://studentsuccess.aua.am/disability-support-services/

Template for Assignment-Specific Rubric*

Assignment Name	Not Yet Competent	Developing	Proficient	Advanced
Homework Tasks	No work submitted; or submitted work is not of the student's. Theoretical questions have vast holes in the arguments presented with inconsistent and incorrect notation. Programming problems are unfinished and hence, do not compile. Documentation is lacking and an efficient solution is not presented.	The work submitted is entirely written by the student. Theoretical questions have many gaps in the arguments presented with incorrect notation. Programming problems do not conform to the specifications and do not hand special or error cases. The program is not well-organized. Documentation is lacking. The problem is not solved in an efficient manner.	The work submitted is entirely written by the student. Theoretical questions have small gaps with inconsistent notation. Programming problems conform to the specification given in the assignment and handle special and error cases. The solution takes into account design and style to be coherent and well-organized. Documentation is lacking. The program is efficient, but not necessarily easy to use.	The work submitted is entirely written by the student. Theoretical questions are answered correctly with no gaps in the arguments and with correct notation. Programming problems conform to the specification given in the assignment and handle special and error cases. The solution takes into account design and style to be coherent and well-organized. Documentation is provided throughout the work with helpful identifiers and clear layout. The program should be efficient and easy to use.

More items will be added during the course.