CPTR 242 Sequential and Parallel Data Structures and Algorithms Spring 2019

Walla Walla University—Seventh-day Adventist Higher Education

Time/Place:	MWRF 10:00–10:50 AM, Kretschmar Hall 348, Walla Walla University
Instructor:	E. Preston Carman Jr., Assistant Professor of Computer Science
E-mail:	preston.carman@wallawalla.edu
Office:	Kretschmar Hall 328, (509) 527-2084
Office Hours:	See course web site.
Text:	Dale et al., *C++ Plus Data Structures,* Jones & Bartlett Learning., 2018. ISBN: 978-1-284-08918-9.
Web:	*https://class.wallawalla.edu/*

Course Description. Introduction to advanced data structures and the algorithms that manipulate them. Students will create and manipulate linked lists, stacks, queues, graphs, trees, and hash tables. Students will also search and sort using various common algorithms, both sequentially and in parallel. Prerequisite: CPTR 142. 4 credits.

Goals. This course promotes the mission of Walla Walla University in general and the Computer Science program in particular through the following goals. Upon completion of this course, students will have:

- **Excellence in thought.** Learn elementary theory and concepts of data structures and relating algorithms. Develop an ability to think critically and confidently in system problem solving.
- **Generosity in service.** Begin to learn the potential of application areas. Contribute to a shared learning environment by discussing topics in-class.
- **Beauty in expression.** Strengthen ability to represent and communicate design and solutions using appropriate data structures and algorithms.
- **Faith in God.** Begin to gain an awareness of ethical and moral issues of computing from a Christian worldview, towards a field for which you have enormous potential to impact society.

Schedule. The following schedule will be adhered to as much as possible but is subject to change based on the professional judgment of the instructor. **Check your D2L account regularly** for all announcements, assignments, lecture slides and grades posted in D2L. **Check your Gitlab account regularly** for all assignments and projects posted in gitlab.

The final exam may be taken out of schedule only with the approval of the Associate Vice President for Academic Administration, Dr. Scott Ligman.

Late Policy. All work is due at 11:59 PM on the due date. Late work is not accepted unless arranged with the instructor prior to the due date. A **Late Penalty** of 10% per day late is applied on homework submitted after the due date. Homework **not** accepted more than 1 week late. Missed tests will receive a score of 0 and may not be made up.

Assessment. All materials submitted by a student will be evaluated in a timely manner (typically 2 weeks). Exams will be scored and be accessible for viewing by a student in a timely manner (typically 1 week). The score for each class requirement may be accessed in the D2L System. The final letter grade will be calculated as follows:

10%	Class Participation	Most days will include an assigned text-book reading and will include a few questions to be answered before class. During class these reading guides will be used for in-class discussion and highlighting specific concepts from the lecture. Students will work in a group to present an in-depth talk on a single topic during the quarter.
20%	Homework	Weekly homework will be turned in on Monday before 11:59 PM.
30%	Programming Projects	Three programming projects will be assigned over the quarter and are due on Wednesday of week 4, 7, and 10.
20%	Quizzes	Every Friday start with a quiz over the content since the last quiz.
20%	Final exam	The final exam will be up of two parts: practical and theory. The practical exam will be a take home programming assignment. The theory will take place during the final exam time slot and include several interview type essay questions.

Some assignments will be more challenging than previous assignments.

Appropriate (to your instructor) modification of the final letter grade may be made in individual cases for progress, unusual circumstances, etc. Around weeks 4 and 7 there will be progress reports for students scoring at or below a C− and for certain other students considered to be at risk.

		B+	89%	C+	78%	D+	67%
A	93%	B	82%	C	71%	D	60%
A−	91%	B−	80%	C−	69%	D−	58%

Workload: According to the Department of Education, a credit hour is an amount of work represented in learning outcomes and verified by student achievement that is not less than one hour of classroom or direct faculty instruction and a minimum of two hours of out-of-class student work each week. I expect you to work 2-3 hours per week outside of class for every hour spent in class. Be sure to set aside this much time from the beginning of the quarter so as not to get behind.

Academic Integrity. Some collaboration on homework is allowed—and even encouraged—but work you submit for grading must be your own. Any type of cheating on an exam, including (but not limited to) copying another student's work or using unauthorized electronics or notes, will result in disciplinary action. See *http://www.wallawalla.edu/academics/academic-administration/academic-policies/academic-integrity-policy/* .

Disabilities. Appropriate accommodations are available for students with documented disabilities. To inquire about accommodations, contact Sue Huett at Disability Support Services (527-2366). This syllabus is available in alternative formats upon request. See
http://www.wallawalla.edu/resources/student-support-services/disability-support-services/ .

Emergency Procedures. Please refer to the web page *http://www.wallawalla.edu/security* .

Table 1: Tentative Schedule for Sequential and Parallel Data Structures and Algorithms (CPTR 242)

W	DATE	TOPIC	READING	DUE
1	Mon 4/1	Syllabus, Arrays and Pointers		
	Wed 4/3	Software Engineering Principles	Ch1	
	Thu 4/4	Data Design and Implementation	Ch2.1-2.3	
	Fri 4/5	Object-Oriented Programming and Comparison	Ch2.4-2.6	Q01
2	Mon 4/8	ADT Unsorted List	Ch3.1-3.2	HW1
	Wed 4/10	Implementing an Unsorted List as a Linked Structure	Ch3.3-3.4	
	Thu 4/11	Comparing Unsorted List Implementations	Ch3.5	
	Fri 4/12	Review and Catch-Up		Q02
3	Mon 4/15	ADT Sorted List	Ch4.1	HW2
	Wed 4/17	Implementing a Sorted List as a Linked Structure	Ch4.2-4.3	
	Thu 4/18	Comparison of Unsorted and Sorted Lists	Ch4.4-4.5	*Talk1*
	Fri 4/19	ADT Stack	Ch5.1-5.2	Q03
4	Mon 4/22	ADT Queue	Ch5.3	HW3
	Wed 4/24	Implementing a Queue as a Linked Structure	Ch5.4	**P1**
	Thu 4/25	Graphs	Ch13	*Talk2*
	Fri 4/26	Review and Catch-Up		Q04
5	Mon 4/29	Graphs, cont.		HW4
	Wed 5/1	List Plus	Ch6.1-6.4	
	Thu 5/2	Copy Structures	Ch6.5-6.6	*Talk3*
	Fri 5/3	Programming with Recursion	Ch7.1-7.4	Q05
6	Mon 5/6	Writing Recursive Functions	Ch7.5-7.6	HW5
	Wed 5/8	Recursive Versions of	Ch7.7-7.9	
	Thu 5/9	How Recursion Works	Ch7.10	*Talk4*
	Fri 5/10	Review and Catch-Up		Q06
7	Mon 5/13	Tracing a Recursive Execution	Ch7.11-7.12	HW6
	Wed 5/15	Debugging and Removing Recursion	Ch7.13-7.15	**P2**
	Thu 5/16	Binary Search Trees	Ch8.1-8.5	*Talk5*
	Fri 5/17	Recursive Binary Search Tree Operations	Ch8.6a	Q07
8	Mon 5/20	Recursive Binary Search Tree Operations, cont.	Ch8.6b	HW7
	Wed 5/22	Iterative Binary Search Tree Operations	Ch8.7-8.8	
	Thu 5/23	Priority Queue	Ch9.1-9.2	*Talk6*
	Fri 5/24	Review and Catch-Up		Q08
9	Mon 5/27	**No Class**		HW8
	Wed 5/29	Heaps	Ch9.3	
	Thu 5/30	Maps and Sets	Ch11.1-11.2	*Talk7*
	Fri 5/31	Hashing	Ch11.3	Q09
10	Mon 6/3	Sorting	Ch12.1-12.4	HW9
	Wed 6/5	Sorting Efficiency	Ch12.5-12.6	**P3**
	Thu 6/6	Parallel Merge Sort	Ch12.8	
	Fri 6/7	Review		Q10
11	Mon 6/10			HW10
	Wed 6/12	**Final Exam at 12:00-1:50 pm**		