

CS2013 Course Syllabus
Fall 2017

Section 1: Lecture: Tue/Thu 6:00 P.M. – 6:50 P.M. E&T 309

Lab: Tue/Thu 6:50 P.M. – 8:05 P.M. E&T 309

Section 2: Lecture Mon/Wed 3:05 P.M. – 3:55 P.M. E&T 210

Lab: Mon/Wed 4:00 P.M. – 5:15 P.M. E&T 210

Instructor	Jungsoo (Sue) Lim, Harminder Singh Email: jlim34@calstatela.edu , minder91singh@gmail.com Office hours: 1. Monday - Thursday 9:00 AM – 10:30 AM at E&T A310 2. Tuesday, Thursday 5:00 PM – 6:00 PM at E&T A310 3. By Appointment
Course name	Programming with Data Structures
Credits	3 units
Contact hours	4 hours/week
Text book	Data Structures & Algorithms by Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser in Java, 6th Edition (ISBN: 978-1-118-77133-4)
Course Information	a) Advanced programming techniques; elementary data structures such as dynamic arrays, linked lists, stacks, queues, and trees, sorting and searching algorithms. Laboratory activities on problem analysis and software development. b) Prerequisites: CS2012 c) Recommended Prerequisite: Math 207, Math 248 d) This course is required in the BS program
Course Goals	The Student Learning Outcomes that are addressed by the course are: <i>SLO #2: Students will be able to demonstrate fluency in at least one programming language and acquaintance with at least three more.</i> <i>SLO #3: Students will have a strong foundation in the design, analysis, and application of many types of algorithms.</i> <i>SLO #5. Students will have the training to analyze problems and identify and define the computing requirements appropriate to their solutions.</i> <i>SLO #6. Students will have the training to design, implement, and evaluate large software systems working both individually and collaboratively.</i> <i>SLO #7. Students will be able to communicate effectively orally and in writing.</i> Other outcomes of instruction: At the end of the course students are able to: <ul style="list-style-type: none">• Use recursion as a tool to solve some specific problems.

Brief list of topics to be covered

- Know the standard Abstract Data Types, and their implementations.
- Study and use different available JAVA Data Structures.
- Know the standard searching and sorting algorithms and their efficiency.
- Understand the complexity analysis for some simple software.
- Recursion
- Algorithm Efficiency (brief)
- Sorting
- Lists, Stacks, Queues, and Priority Queues
- Binary Search Trees
- Graphs and Applications (Optional)

Laboratory Projects

Each week students will complete a 3-hour lab projects on selected topics, except the exam weeks. At the end of each lab students will turn in lab programming assignment.

Out of class Assignments

Each week students will have an assignment due on the following week, except the exam weeks. For these assignments students may required to complete an unfinished implementation, design/implement a system, produce system document (pseudo code, UML, etc), and prepare a user's manual.

Quizzes

Each class will have in-class quiz.

Grading Policy

Quizzes: 10%
Laboratory assignments: 20 %
Out of class assignments: 10%
2 Midterm: 30%
Final: 30%

A-, A	90 – 100
B-, B, B+	80 – 89
C-,C,C+	65 – 79
D,F	<65

Academic Integrity

Cheating will not be tolerated. Cheating on any assignment or exam will be taken seriously. All parties involved will receive a grade of F for the course and are reported to the proper authorities.

ADA Statement

Reasonable accommodation will be provided to any student who is registered With the Office of Students with Disabilities and requests needed accommodation.

CS2013 Course Weekly Schedule

Week	Lecture topics	Homework
1	Review <ul style="list-style-type: none"> • Encapsulation – data protection • Inheritance – super class/sub class, constructor chaining, overriding vs. overloading • Polymorphism – dynamic binding, declared type, actual type, casting (implicit casting/explicit casting, up casting/down casting) • Generics – creating generic class and method, benefits of generics, and restrictions • Java FX – creating application, inserting components, adding event handlers, and adding UI controls. 	R-2.1 – R-2.15
2	Chapter 3 – Fundamental Data Structure <ul style="list-style-type: none"> • Array List • Linked List <ul style="list-style-type: none"> ○ Singly linked list ○ Circularly linked list ○ Doubly linked list • Array List vs. Linked List 	R-3.1- R-3.16 C-3.17- C-3.28
3	Chapter 4 – Algorithm Analysis <ul style="list-style-type: none"> • Compare growth rates • Estimate algorithm efficiency using the Big O notation • Examples of algorithm analysis 	R-4.1- R-4.28 C-4.35 - C-4.46
4	Chapter 5 – Recursion <ul style="list-style-type: none"> • Review what a recursion method is and its benefits • Examples of recursions • Analyze recursive algorithms • Design recursive algorithms • Eliminating tail recursion 	R-5.1- R-5.10 C-5.11 - C-5.19
5	Chapter 6 – Stacks, Queues, and Double-Ended Queues (Deque) <ul style="list-style-type: none"> • Stacks – ADT and Implementation • Queues – ADT and Implementation 	R-6.1- R-6.15 C-6.16 - C-6.22

	<ul style="list-style-type: none"> • Deques – ADT and implementation 	
6	Chapter 7 – List and Iterator ADTs <ul style="list-style-type: none"> • Array Lists • Positional Lists • Iterators • Java Collection Framework <ul style="list-style-type: none"> ○ Overview of Lists Iterators ○ Overview of List-based Algorithms Midterm 1 (Chapter 3 – Chapter 7)	R-7.1- R-7.24
7	Chapter 8 – Trees <ul style="list-style-type: none"> • General Trees • Tree ADT • Binary Trees • Implementing Trees • Tree Traversal (in-order, pre-order, post-order, breath first traversal) 	R-8.1- R-8.20 C-8.27 - C-8.30 C-8.52
8	Chapter 9 – Priority Queues <ul style="list-style-type: none"> • Priority Queue ADT • Implementing Priority Queue • Heaps • Sorting with a Priority Queue <ul style="list-style-type: none"> ○ Selection sort ○ Insertion sort ○ Heap sort 	R-9.1- R-9.24 C-9.25 - C-9.34 C-9.43
9	Chapter 10 – Maps, Hash tables, and Skip Lists <ul style="list-style-type: none"> • Maps ADT • Hash tables – hash function, collision-handling, load factors, rehashing, and efficiency • Sorted Maps • Skip Lists • Sets, Multisets, and Multimaps 	R-10.1- R-10.20 C-10.34 - C-10.35
10	Chapter 11 – Search Trees <ul style="list-style-type: none"> • Binary Search Trees (BST) 	R-11.1- R-11.24

	<ul style="list-style-type: none"> Balanced search trees <ul style="list-style-type: none"> AVL Tree Splay trees (2,4) Trees Red-black Trees <p>Midterm 2 (Chapter 8 – Chapter 11)</p>	C-11.28 - C-11.48
11	<p>Chapter 12 – Sorting and Selection</p> <ul style="list-style-type: none"> Merge Sort Quick Sort Selection <ul style="list-style-type: none"> Prune-and-search Randomized quick-select 	R-12.1- R-12.18 C-12.26 - C-12.29
12	<p>Chapter 14 – Graphs Algorithm - I</p> <ul style="list-style-type: none"> Data Structure for Graphs Graph Traversals Shortest Paths <ul style="list-style-type: none"> Weighted Graphs Dijkstra’s Algorithm 	R-14.1- R-14.36
13	<p>Chapter 14 – Graphs Algorithm - II</p> <ul style="list-style-type: none"> Minimum Spanning Tree <ul style="list-style-type: none"> Prim’s Algorithm Kruskal’s Algorithm 	R-14.1- R-14.36
14	<p>Chapter 15 – Memory Management and B-Tree</p> <ul style="list-style-type: none"> Memory Management Memory Hierarchies and Caching (a, b) Trees B-Tree 	R-15.1- R-15.8
15	Review for final exam	
16	Final Exam	