CSC 241
DATA STRUCTURES
Spring 2016, TR 11:00 – 12:15 p.m., ED 324
(936) 468-2508

Dr. Robert Strader 304e McKibben rstrader@sfasu.edu

PREREQUISITE: CSC 202; 211 highly recommended

OFFICE HOURS: 10:50 – 11:50 a.m., 1:30 – 2:30 p.m. MW;
10:00 – 11:00 a.m., 12:30 – 1:30 p.m. TTh.
By Appointment M-F.

REQUIRED MATERIALS FOR CSC 241:

Data Abstraction and Problem Solving with Java, 3rd Ed., Prichard and Carrano,
Addison-Wesley, 2011.

OTHER MATERIALS:

Any Java textbook: (i.e., Intro. to Java Programming, 9th/10th Ed., Liang, Prentice Hall.)
Calculator – “four function” for exams.

EXAMINATIONS: (67% of the course grade)
2 Class Examinations (first - 100 points, second - 100 points)
Final Examination -- Comprehensive (200 points)

Note: There are no exemptions from the final examination. Check the final exam time.
If the final exam time (05/10/16 10:30 a.m. – 12:30 p.m.) is a problem, you need
to drop this course.

ASSIGNMENTS: (33% of the course grade)
Programming assignments (200 pts)

Attendance and class participation - expected

LITERATURE:

IEEE/CS Computer TK 7885 A1 I6
ACM Communications QA 76 A772

GRADING: <60 F, 60-69 D, 70-79 C, 80-89 B, >89 A; adjusted for difficulty
Attendance & Class Behavior: Roll will be taken regularly. Attendance will not be taken into consideration for your final grade. If you are absent from class, please do not come by my office and ask me to repeat the class lecture. There will be no smoking, no chewing of tobacco, no eating or drinking, no bare feet, and no wearing of hats during class. Please keep your feet off of the seat backs and seats. No disruptive behavior including offensive language will be tolerated in a computer science facility or related activity. Such behavior may result in administrative removal from class. Cell phones and pagers should be turned off for the duration of class. Surfing, texting, or other distracting behavior should not be initiated. Only students officially registered for the course and approved assistants may attend class.

Examination Policy: All class examinations are considered to be a major part of the course work upon which a large part of the course grade depends. There are NO make-up exams! Class examinations will be announced at least two classes prior to the examination. If you have a conflict with another university event, you must contact me well in advance of the examination. In case of an extreme emergency, contact me before the scheduled examination. Failure to do so may result in an examination grade of zero.

Assignment Policy: All assignments are due at the BEGINNING of class (or by an announced time) on the specified due date. That means any assignment given to me after I have collected the assignments from the class is considered to be late. There will be NO late assignments accepted without an appropriate reason (doctor's note, university event - see above, etc.). Under NO circumstances will any assignment be accepted for credit after the collected class assignments have been graded and returned. DO NOT place assignments in my box or under my office door during class. If you have a conflict, please contact me in advance. PLEASE NOTE: You may be given assignments during the last five class days of the semester.

Software Policy: Disciplinary action will be taken against individuals who perform unauthorized duplication of software or who are involved in the unauthorized use of duplicated software. Such action may make it impossible for you to successfully complete this course.

Computer Laboratory Usage: Students utilizing equipment in university computing laboratories are expected to read and abide by all posted policies for the laboratories. Please note that no children and no pets are permitted in university computing laboratories.

Drop Policy (Univ.): The date (03/28/16) is the last day to drop this course with a W.

Special Accommodation Requests: Students with special accommodation requests have the responsibility to immediately initiate a meeting with the instructor to discuss how the special accommodations will be provided. Students who are aware of these special needs at the beginning of the semester must inform the instructor in person before the twelfth class day about any class activity, which will require special accommodations.

Computer Account Policy: All assignments that require the use of the University Computer must be done under the computer account that is assigned to you in this class. You should NOT do other class assignments in this account, and you should NOT do assignments from this class in other accounts. Failure to abide by university and departmental computer account policies could mean receiving a grade of F in this course.

Cheating Policy: If in my judgment a student is found cheating on an examination, a grade of zero will be assigned as the examination grade and a minimum of one (1) letter grade will be lost in the course grade. A course grade of F may be assigned depending on the situation. A student found cheating on an examination may not drop the course.

All class assignments are to be done INDEPENDENTLY unless part of a team project. If in my judgment two or more people hand in (non-team) assignments that I judge to be the same, a grade of zero may be awarded to all involved assignments and a minimum of one letter grade may be lost in the course grade. A recurrence of this by any individual will result in a grade of F in the course. Students should save all developmental copies of their programs so that individual program development can be verified by me if I think it is necessary. DO YOUR OWN WORK!!!!! Do NOT show other students your code!!!
**Identification**: Valid student I.D. cards must be presented on each examination day. (No I.D...No exam...Grade of zero)

The following web pages contain pertinent information:

- The program learning outcomes for this course can be found at: [http://cs.sfasu.edu/cs/plo/](http://cs.sfasu.edu/cs/plo/)
- General student policies and information can be found at:
  - University Academic Integrity policy: [http://www.sfasu.edu/policies/academic_integrity.asp](http://www.sfasu.edu/policies/academic_integrity.asp)
  - University Withheld Grades policy: [http://www.sfasu.edu/policies/course-grades.pdf](http://www.sfasu.edu/policies/course-grades.pdf)
  - Students with disabilities information: [http://www.sfasu.edu/policies/academic-accommodation-for-students-with-disabilities.pdf](http://www.sfasu.edu/policies/academic-accommodation-for-students-with-disabilities.pdf)

**Acceptable Student Behavior**: Classroom behavior should not interfere with the instructor’s ability to conduct the class or the ability of other students to learn from the instructional program (see the Student Conduct Code, policy D-34.1). Unacceptable or disruptive behavior will not be tolerated. Students who disrupt the learning environment may be asked to leave class and may be subject to judicial, academic or other penalties. This prohibition applies to all instructional forums, including electronic, classroom, labs, discussion groups, field trips, etc. The instructor shall have full discretion over what behavior is appropriate/inappropriate in the classroom. Students who do not attend class regularly or who perform poorly on class projects/exams may be referred to the Early Alert Program. This program provides students with recommendations for resources or other assistance that is available to help SFA students succeed.

CSC 241 - DATA STRUCTURES

**CREDIT HOURS**: 3

**PREREQUISITES**: CSC 202; CSC 211 recommended

**GRADE REMINDER**: Must have a grade of C or better in each prerequisite course.

**CATALOG DESCRIPTION**

Advanced programming techniques including indirection and recursion. Conceptual development and implementation of data structures including arrays, records, linear lists, stacks, queues, trees, tables, and graphs. Applications involving strings, sorting, searching, and file operations.

**PURPOSE OF COURSE**

The purpose of this course is to familiarize the student with advanced programming techniques and to introduce the student to the most commonly used methods of data organization. Emphasis is placed on advanced programming concepts and use of information structures in applications from both physical and logical views.

**EDUCATIONAL OBJECTIVES**

Upon successful completion of the course, students should be able to:

1. Demonstrate knowledge of the software life cycle and the program development process.
2. Analyze problems and develop program designs with a variety of data structures including stacks, queues, lists, strings, tables, trees and graphs involving both definition and implementation issues.
3. Apply analysis techniques to problems involving iteration and recursion.
4. Create small program systems from carefully specified requirements using software
engineering design and reuse principles, appropriate data structure designs, and algorithmic and program performance measures.

5. Describe well known problems and solutions in computation including searching, sorting, arithmetic evaluation, backtracking, programming languages, and string manipulation.

6. Develop and implement abstract data type specifications.

7. Apply comprehensive language features including indirection.

8. Develop both structured procedural and object oriented solutions.

9. Demonstrate an understanding of machine memory organization and operation.

CONTENT

**CONTENT**

**Hours**

Programming Concepts Review .................................................................6
  Specification, design and implementation issues
  Abstract data type (ADT) concept
  Scalars, arrays, records, sets, files
  Correctness and analysis

Advanced Programming Techniques ..........................................................6
  Indirection
  Recursion
  Examples: strings, linked lists

Linear Data Structures .................................................................................8
  Linear lists (array, bit, and linked representations)
  Stacks
  Queues
  Examples: infix to postfix, postfix evaluation, memory allocation

Trees ............................................................................................................6
  Binary trees
  General trees
  Examples: traversal algorithms

Sorting ............................................................................................................3
  Selection sorts, insertion sorts, exchange sorts including both \( N^2 \) and \( N \log N \) sorts.
  Examples: Shell, quick, radix, bubble, tree, heap, merge

Searching ......................................................................................................3
  Sequential, binary, hashing
  Example: traversal

Files and Directories ....................................................................................4
  Sequential, indexed, direct, inverted, chained
  Example: file merge

Advanced Topics ..........................................................................................6
  Graphs and digraphs
  Matrix and list representations
Examples: traversal, minimum spanning tree
Special trees
B, B*, B+, AVL, Red-Black
Examples: traversal, dynamic data organization

Exams............................................................................................................................................. 3
TOTAL 45

REFERENCES

CSC 241 - Spring 2016
Tentative Schedule

<table>
<thead>
<tr>
<th>Date</th>
<th>Day</th>
<th>Topic/Reading</th>
<th>Ch</th>
<th>Assignment Completion</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/19</td>
<td>T</td>
<td>Introduction and Overview, Design, UML</td>
<td>Ch 2</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Th</td>
<td>Recursion and Examples, UNIX, UNIX</td>
<td>Ch 3</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>T</td>
<td>ADTs, Classes, Interf., Package, lists</td>
<td>Ch 4</td>
<td>(Hw 1)</td>
</tr>
<tr>
<td>28</td>
<td>Th</td>
<td>Classes – Linked lists, Refs, Dynamic All.</td>
<td>Ch 4/Ch 5</td>
<td>Lab 1</td>
</tr>
<tr>
<td>2/02</td>
<td>T</td>
<td>Linked Lists: def., ops, array vs. ref.</td>
<td>Ch 5</td>
<td>(Hw 2)</td>
</tr>
<tr>
<td>04</td>
<td>Th</td>
<td>Linked Lists: Files, Circular, Double, JFC</td>
<td>Ch 5</td>
<td></td>
</tr>
<tr>
<td>09</td>
<td>T</td>
<td>Recursion: Applications - 8Q, langs</td>
<td>Ch 6</td>
<td>Lab 2</td>
</tr>
<tr>
<td>11</td>
<td>Th</td>
<td>Recursion: Induc./Review</td>
<td>Ch 6</td>
<td>(Hw 3)</td>
</tr>
<tr>
<td>16</td>
<td>T</td>
<td>Test 1</td>
<td>(Ch 2 – Ch 6)</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Th</td>
<td>Stacks: ADT, Impl., PFEval</td>
<td>Ch 7</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>T</td>
<td>Stacks: Appl., AE, Search</td>
<td>Ch 7</td>
<td>Lab 3</td>
</tr>
<tr>
<td>25</td>
<td>Th</td>
<td>Queues: Intro, Impl., Apps, Simulation</td>
<td>Ch 8</td>
<td></td>
</tr>
<tr>
<td>3/01</td>
<td>T</td>
<td>Java: Inheritance, access</td>
<td>Ch 9</td>
<td>(Hw 4)</td>
</tr>
<tr>
<td>03</td>
<td>Th</td>
<td>Java: Dynamic Binding, Abstract, Lists</td>
<td>Ch 9</td>
<td></td>
</tr>
<tr>
<td>08</td>
<td>T</td>
<td>Java: Generics, Iterators, JFC, Sorting</td>
<td>Ch 9/Ch 10</td>
<td>(Hw 5)</td>
</tr>
<tr>
<td>10</td>
<td>Th</td>
<td>Sorting/Big Oh – sel., inser., exchange</td>
<td>Ch 10</td>
<td>Lab 4</td>
</tr>
<tr>
<td>15</td>
<td>T</td>
<td>----------------------------------------</td>
<td>----------------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>17</td>
<td>Th</td>
<td>----------------------------------------</td>
<td>----------------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>22</td>
<td>T</td>
<td>Sorting: Merge, Quick, Radix/Trees: Intro</td>
<td>Ch 10/Ch 11</td>
<td>(Hw 6)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>---</td>
<td>-----------------------------------------------------------------</td>
<td>----</td>
<td>-----------------------------------------------------------------</td>
</tr>
<tr>
<td>23</td>
<td>W</td>
<td>----------------------------- W Drop -----------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Th</td>
<td>Easter Break</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>T</td>
<td>Trees: Intro., Reps, ADT Binary tree Ch 11 Lab 5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>Th</td>
<td>Trees – ADT BST, Eff., Tree Sort, Files Ch 11 (Hw 7)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4/05</td>
<td>T</td>
<td>Test 2 (Ch7 – Ch 11)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>07</td>
<td>Th</td>
<td>Tables: Intro., Impl., Ch 12 Lab6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>T</td>
<td>Tables: PQ, Heaps, Heapsort Ch 12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>T</td>
<td>Adv. Trees: Balanced – AVL, B, RB Ch 13/15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Th</td>
<td>Adv. Trees: 2-3, 2-3-4, B/external Ch 13/15 Lab 7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>T</td>
<td>Graphs: Introduction, ADT, Reps Ch 14 (Hw 9)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>Th</td>
<td>Graphs: Traversals, MST, Apps Ch 14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5/03</td>
<td>T</td>
<td>Graphs: Application/Theory Ch 14 (Hw 10)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>05</td>
<td>Th</td>
<td>SE, Advanced Courses/Review for Final All Lab 8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>T</td>
<td>Final Exam (10:30 a.m. – 12:30 p.m.) All</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>