Shaw University
College of Graduate and Professional Studies
Department of Computer Information Sciences

Course Number: CSC330 Fall 2008
Course Name: Data Structures and Algorithms
Credit Hours: 3

Instructor’s Name: Dr. Wei Jin
Office Location and Number: Graphics Building#7
Classroom & Time: Science Building 301;
               MWF 1pm-1:50pm
Phone Number: (919) 546-8376
E-mail Address: wjin@shawu.edu
Office Hours:
               MW:  11am – 12:00pm
               TuTh: 12:30pm – 3:30pm
               Fri.: 11am – 1pm

Shaw University Help Desk: 919-546-8587, helpdesk@shawu.edu

COMPUTER SCIENCE PROGRAM MISSIONS, GOALS AND LEARNING OUTCOMES

Mission Statement
The mission of the Bachelor of Science in Computer Science degree Program is to provide in-depth study of the fundamentals of computer science and important current issues as well as develops students’ analytical and problem-solving skills. We also attempt to instill in them the attitudes and values that will prepare them for a lifetime of continued learning and leadership. We aim to help students obtain productive employment or pursue advanced degrees in computer science and related technology fields.

Computer Science Program Goals
1. Increasing retention of Computer Science major students.
2. Graduating Computer Science students who are competent in their discipline.

Computer Science Program Learning Outcomes (PLOs)

1. Problem Solving and Critical Thinking in Computational Practice: Solve abstract and complex problems using software design methodology. Make informed choices among alternative solutions. The student will be able to:
   b. Implement an algorithm by creating a tested and debugged programmatic solution.
c. Examine and analyze alternative solutions to a problem.
d. Develop abstract models to simulate complex systems.
e. Determine correctness and efficiency of a system design and implementation.

2. **Knowledge of Advanced Computing Topics:** Demonstrate understanding of the principles and current technologies in computer architecture, operating systems, computer networks, database systems, programming languages and compilers.

3. **Communication and Interpersonal Skills:** Use written, oral and electronic methods for effective communication. The student will be able to:
   a. Document all aspects of a system precisely and clearly.
   b. Use written, oral, and electronic communication to convey technical information effectively.
   c. Devise effective user interfaces.
   d. Work cooperatively in teams and with others.

4. **Ethical and Professional Responsibilities:** Discern and articulate the impact of technologies on society. The student will be able to:
   a. Plan for and ensure the security, privacy, and integrity of data.
   b. Recognize the ethical, legal, and social implications of computing.
   c. Demonstrate an understanding of the Association of Computing Machinery (ACM) Code of Professional Ethics.
   d. Analyze the impact that computing has on the global society.
   e. Recognize the need for continuing professional development.

**TEXT**

**Required Textbooks:** Problem Solving, Abstraction, and Design Using C++ (4th or 5th Edition) by Frank L. Friedman  Elliot B. Koffman. Publisher: Addison Wesley

**Supplementary Materials:** Lecture notes and online exercises can be found on Blackboard under the course named “Data Structures and Algorithms”

**Hardware Requirements:** A PC with MS Windows Operating System

**Software Requirements:** Borland C++ and Internet Explore

**COURSE DISCRIPTION:**

**Prerequisites and Corequisites:** CSC202.

**Course Description:** This course builds on the foundation provided by the CSC201 and CSC202 sequence to introduce the fundamental concepts of data structures and the algorithms that proceed from them. Topics include recursion, the underlying philosophy of object-oriented
programming, fundamental data structures (including stacks, queues, linked lists, hash tables, trees, and graphs), and the basics of algorithmic analysis.

Topics:

- Review of elementary programming concepts
- Fundamental data structures: Stacks; queues; linked lists; hash tables; trees; graphs
- Object-oriented programming: Object-oriented design; encapsulation and information hiding; classes; separation of behavior and implementation; class hierarchies; inheritance; polymorphism
- Fundamental computing algorithms: O(N log N) sorting algorithms; hash tables, including collision-avoidance strategies; binary search trees; representations of graphs; depth- and breadth-first traversals
- Recursion: The concept of recursion; recursive mathematical functions; simple recursive procedures; divide-and-conquer strategies; recursive backtracking; implementation of recursion
- Basic algorithmic analysis: Asymptotic analysis of upper and average complexity bounds; identifying differences among best, average, and worst case behaviors; big "O," little "o," omega, and theta notation; standard complexity classes; empirical measurements of performance; time and space tradeoffs in algorithms; using recurrence relations to analyze recursive algorithms
- Algorithmic strategies: Brute-force algorithms; greedy algorithms; divide-and-conquer; backtracking; branch-and-bound; heuristics; pattern matching and string/text algorithms; numerical approximation algorithms
- Software engineering: Software validation; testing fundamentals, including test plan creation and test case generation; object-oriented testing

Student Learning Outcomes (SLOs): After taking this course, students are expected to be able to

**Fundamental data structures:**

1. Describe common applications for each data structure in the topic list.
2. Implement the user-defined data structures in a high-level language.
3. Compare alternative implementations of data structures with respect to performance.
4. Write programs that use each of the following data structures: arrays, records, strings, linked lists, stacks, queues, and hash tables.
5. Compare and contrast the costs and benefits of dynamic and static data structure implementations.
6. Choose the appropriate data structure for modeling a given problem.

**Recursion:**

7. Describe the concept of recursion and give examples of its use.
8. Describe how recursion can be implemented using a stack.
9. Compare iterative and recursive solutions for elementary problems such as factorial.
10. Describe the divide-and-conquer approach.
11. Implement, test, and debug simple recursive functions and procedures.
12. Determine when a recursive solution is appropriate for a problem.

**Basic Algorithm Analysis:**

13. Explain the use of big O, omega, and theta notation to describe the amount of work done by an algorithm.
14. Use big O, omega, and theta notation to give asymptotic upper, lower, and tight bounds on time and space complexity of algorithms.
15. Deduce recurrence relations that describe the time complexity of recursively defined algorithms.
16. Solve elementary recurrence relations.

**Algorithmic strategies:**

17. Describe the shortcoming of brute-force algorithms.
18. For each of several kinds of algorithm (brute force, greedy, divide-and-conquer, backtracking, branch-and-bound, and heuristic), identify an example of everyday human behavior that exemplifies the basic concept.
19. Implement a greedy algorithm to solve an appropriate problem.
20. Implement a divide-and-conquer algorithm to solve an appropriate problem.
21. Use backtracking to solve a problem such as navigating a maze.
22. Describe various heuristic problem-solving methods.
23. Use numerical approximation to solve mathematical problems, such as finding the roots of a polynomial.

**Object-oriented programming:**

24. Justify the philosophy of object-oriented design and the concepts of encapsulation, abstraction, inheritance, and polymorphism.
26. Describe how the class mechanism supports encapsulation and information hiding.
27. Design, implement, and test the implementation of "is-a" relationships among objects using a class hierarchy and inheritance.
28. Compare and contrast the notions of overloading and overriding methods in an object-oriented language.
29. Explain the relationship between the static structure of the class and the dynamic structure of the instances of the class.
30. Describe how iterators access the elements of a container.
### Student Learning Outcomes (SLOs)

<table>
<thead>
<tr>
<th>Student Learning Outcomes (SLOs)</th>
<th>Assessment of Student Learning Outcomes (Accessment Tools)</th>
<th>Linkage to Program Learning Outcomes (PLOs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,3,5</td>
<td>Exam Questions</td>
<td>1</td>
</tr>
<tr>
<td>2,4,6</td>
<td>Exam Questions and Programming Projects</td>
<td>1</td>
</tr>
<tr>
<td>7,8,9,10,12</td>
<td>Exam Questions</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>Exam Questions and Programming Projects</td>
<td>1</td>
</tr>
<tr>
<td>13-16</td>
<td>Exam Questions</td>
<td>1</td>
</tr>
<tr>
<td>17,18,22</td>
<td>Exam</td>
<td>1</td>
</tr>
<tr>
<td>19-21,23</td>
<td>Exam Questions and Programming Projects</td>
<td>1</td>
</tr>
<tr>
<td>24, 26, 28-30</td>
<td>Exam Questions</td>
<td>1</td>
</tr>
</tbody>
</table>

### Class Participation:
Class participation is required and you are expected to communicate with other students on team projects, learn how to navigate in Bb, and keep abreast of course announcement. Use the assigned college or university e-mail address as opposed to personal e-mail address. Address technical problems immediately by contacting the HELP desk (919)-546-8587 and observe course netiquette at all times. For example: Always include a subject line before making a comment because; remember without facial expressions some comments may be taken the wrong way. Be careful in wording your emails, the use of emoticons might be helpful in some cases.

### Attendance and Punctuality:
Attendance and punctuality will be weighed in determining your final grade. Students are expected to attend all class discussion group work and to be on time for the class discussion group. If a student is late to class (15 minutes or more) or leaves class early (15 minutes or more) or missing class, it will have the following adverse effect on his/her grade:

- Grade lowered by one grade if 3 times absent or 6 times late (or leaving class early).
- Student will be withdrawn from the course, or take a grade of "F," if 5 or more times absent or 8 or more times late (or leaving class early).

Students absent the week before a quiz or assignment will be expected to take the quiz with the class. Students will be expected to make up any work they may have missed because of their absence or tardiness.

### Grade Evaluation:

<table>
<thead>
<tr>
<th>Grade Evaluation</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework Assignments</td>
<td>40%</td>
</tr>
<tr>
<td>Announced Quizzed and Exams</td>
<td>20%</td>
</tr>
<tr>
<td>Comprehensive Final Exam</td>
<td>30%</td>
</tr>
<tr>
<td>Attendance and Unannounced Quizzes</td>
<td>10%</td>
</tr>
</tbody>
</table>
Course Schedule (Lectures, Assignments, and Exams):

<table>
<thead>
<tr>
<th>Week No.</th>
<th>Topics</th>
<th>Assignments &amp; Exams</th>
<th>SLOs</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (8/21)</td>
<td>Introduction</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 (8/25)</td>
<td>Recursion</td>
<td>Assignment 1</td>
<td>7-12</td>
</tr>
<tr>
<td>2 (9/2)</td>
<td>C++ Arrays and Structures</td>
<td>Assignment 2</td>
<td>4</td>
</tr>
<tr>
<td>3 (9/8)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 (9/15)</td>
<td>C++ Class</td>
<td></td>
<td>4, 24-30</td>
</tr>
<tr>
<td>5 (9/22)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 (9/29)</td>
<td>Mathematics Background for Algorithm Analysis</td>
<td>Assignment 3,4</td>
<td>13-16</td>
</tr>
<tr>
<td>7 (10/6)</td>
<td>Algorithm Analysis</td>
<td>Midterm Grade</td>
<td>13-16</td>
</tr>
<tr>
<td>8 (10/13)</td>
<td>Lists</td>
<td>Assignment 5</td>
<td>1-6, 17-30</td>
</tr>
<tr>
<td>9 (10/20)</td>
<td>Stacks</td>
<td>Assignment 6</td>
<td>1-6, 17-30</td>
</tr>
<tr>
<td>10 (10/27)</td>
<td>Queues</td>
<td>Assignment 7</td>
<td>1-6, 17-30</td>
</tr>
<tr>
<td>11 (11/3)</td>
<td>Binary Trees</td>
<td>Assignment 8</td>
<td>1-6, 17-30</td>
</tr>
<tr>
<td>12 (11/10)</td>
<td>Hashing</td>
<td>Assignment 9</td>
<td>1-6, 17-30</td>
</tr>
<tr>
<td>13 (11/17)</td>
<td>Binary Heap</td>
<td>Assignment 10</td>
<td>1-6, 17-30</td>
</tr>
<tr>
<td>14 (11/24)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15 (12/1)</td>
<td>Final Review</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16 (12/8)</td>
<td>Final Exam</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: Assignments are always due the Friday of the following week after it is assigned. Late submission will incur 20% penalty each week. There will be a maximum of two-week extension for each assignment.

PLAGIARIZING/CHEATING:

Plagiarizing (using another’s work as your own, whether you change variable names or keep it in the original names) and cheating are serious offenses and will be cheated as such. A student who plagiarizes or cheats --- whether giving or receiving information --- will receive a grade of zero on that particular exam or assignment.

LIBRARY:

It is imperative that you familiarize yourself with the instructional materials that are available to you in the James E. Cheek library on campus. Not only are there numerous books, periodicals, magazine articles, encyclopedias, and newspapers on hand for your perusal in our Library. There are course textbooks and related instructional materials that your professors have placed On Reserve for you as well. In addition, there you will have access to state-of-the-art computers and laptops, which help to place the world virtually at your fingertips. Don’t wait another day.

Become a regular patron at the James E. Cheek Library here on the campus of Shaw University. You’ll become a far better scholar, a more capable and well-rounded intellectual, and a sharper and more competitive individual. It’s your library. Use it!
STUDENT CLASSROOM DECORUM EXPECTATIONS:

To enhance the learning atmosphere of the classroom, students are expected to dress and behave in a fashion conducive to learning in the classroom. More specifically, students will refrain from disruptive classroom behavior (e.g., talking to classmates on cell phones, Ipods or similar electronic devices; disrespectful responses to teacher instructions; swearing; wearing clothes that distract from academic learning such as, but not limited to, wearing body-revealing clothing and excessively baggy pants; hats/caps, and/or headdress). Students who exhibit the behaviors described above, or similar behaviors will be immediately dismissed from class on the occurrence of the third documented offense. The student will be readmitted to class only following a decision by the department chair. The student may appeal the decision of the department chair to the Dean of the College offering the course, and, subsequently, to the Office of the Vice President for Academic Affairs, and then to the President of Shaw University. The decision of the President will be final. Failure to follow the procedures herein outlined will result in termination of the appeal, and revert to the decision of the department chair.

Each behavior construed by the teacher/professor as not contributing to learning will be recorded, properly documented, and appropriately reported to the student and to the chair of the academic department offering the course. The report will be in written form with a copy provided to both the student and the department chair. The faculty member should retain a copy for his/her own records.

Additional student behavior codes may be found in Student Affairs especially in the Shaw University Student Handbook.

REFERENCES

- The ACM Code of Ethics (http://www.acm.org/about/code-of-ethics)