

**Fayetteville State University**  
**College of Arts and Sciences**  
**Department of Mathematics and Computer Science**  
**CSC 220.D1 – Data Structures and Algorithms (in Python)**  
**Fall 2011**

**I. Locator Information**

Instructor: Dr. Michael Almeida

Office Location: 340 SBE

Office Phone: 672-1161

Email address: [malmeida@uncfsu.edu](mailto:malmeida@uncfsu.edu)

Office Hours: MW 10:00 – 12:00 & 2:00 – 4:00 & by appointment

Course # and Name: CSC 220 – Data Structures and Algorithms

Day and Time Class Meets: online

Room: online

Semester Credit Hours: 3

Total Contact Hours for Class: 45

**FSU Policy on Electronic Mail:** Fayetteville State University provides to each student, free of charge, an electronic mail account ([username@uncfsu.edu](mailto:username@uncfsu.edu)) that is easily accessible via the Internet. The university has established FSU email as the primary mode of correspondence between university officials and enrolled students. Inquiries and requests from students pertaining to academic records, grades, bills, financial aid, and other matters of a confidential nature must be submitted via FSU email. Inquiries or requests from personal email accounts are not assured a response. The university maintains open-use computer laboratories throughout the campus that can be used to access electronic mail.

Rules and regulations governing the use of FSU email may be found at  
<http://www.uncfsu.edu/PDFs/EmailPolicyFinal.pdf>

**II. Course Description**

This course explores data structures from various viewpoints: data structure design in response to a specific need, the expression of an algorithm in terms of the operations on the data structure, and complexity of operations. This course reinforces the knowledge of data structures gained in CSC130 and extends it, particularly with regard to complexity of corresponding algorithms.

Prerequisites: Grade of “C” or better in CSC 130 and MATH 150.

**III. Disabled Student Services**

In accordance with Section 504 of the 1973 Rehabilitation Act and the Americans with Disabilities Act (ACA) of 1990, if you have a disability or think you have a disability to please contact the Center for Personal Development in the Spaulding Building, Room 155 (1<sup>st</sup> Floor); 910-672-1203.

**IV. Textbook**

Bradley N Miller & David L. Ranum. *Problem Solving with Algorithms and Data Structures Using Python, 1st Edition*. ISBN: 1-59028-053-9. Franklin, Beedle & Associates, 2006.

New and used copies of the textbook are available at the bookstore and on the internet. The 2nd edition of this book has recently been released, but we will be using the 1st edition.

You will also need access to a recent version of Python and an IDE. These are widely available in the computer labs on campus, but you can download and install free copies on your own machines from the following website: <http://www.python.org/download/>

## V. Student Learning Outcomes

Upon completion of this course, students will be able to:

1. implement abstract data types using arrays, lists, and linked structures;
2. implement and use standard data structures such as stacks, queues, deques, heaps, lists, trees and graphs;
3. analyze the complexity of algorithms using asymptotic notations;
4. understand and develop recursive algorithms;
5. implement and use advanced search and sort methods.

## VI. Course Requirements and Evaluation Criteria

a. Final grades are assigned as follows:

A: 90-100

B: 80-89

C: 70-79

D: 60-69

F: < 60

- b. Attendance Requirements – Students are expected to keep up with the readings and the assignments.
- c. The course grade is based on: 6 programming projects (60%) + 2 written exercises (20%) + midterm exam (10%) + final exam (10%)
- d. Policy on Late Assignments - Each assignment must be submitted on time. Five points of your assignment mark will be deducted for each day the assignment is overdue. No assignments will be accepted after one week without a good excuse. If you need to hand in an assignment late, please notify me ahead of time.
- e. Dishonesty on graded assignments will not be tolerated. Although students may discuss assignments with each another, they must neither give nor receive excessive help. Students learn by doing things themselves. Having access to another student's work on the system is definitely not allowed. Duplicate answers are not acceptable. Each student is responsible for disposing of printouts safely (Do NOT simply throw away printouts in a trash can where they can easily be retrieved by another person.) and for protecting their home directory. All students involved in dishonesty (those giving as well as those receiving unallowable help) will be penalized.

Please note: If these evaluation criteria must be revised because of extraordinary circumstances, the instructor will distribute a written amendment to the syllabus.

**VII. Academic Support Resources** – none for this course.

## VIII. Course Outline and Assignment Schedule\*

week #	start date	topics	assignments	events
1	18-Aug	Introduction	Download & install Python	Classes start 8/18
2	22-Aug	Python Lists	Project #1	
3	29-Aug	Python Classes & Objects	Project #2	
4	5-Sep	Ch.2 Stacks & Queues	Project #3	Holiday 9/5
5	12-Sep	Ch.2 Deques		

6	19-Sep	Ch.4 Algorithm Analysis	Written Exercise #1	Fall Convocation 9/15
7	26-Sep	Ch.4 Algorithm Analysis (cont)		
8	3-Oct		Midterm Exam assigned	Midterms start 10/6
9	10-Oct		Midterm Exam due	Homecoming 10/15
10	17-Oct	Ch.4 Searching & Hashing	Project #4	Fall Break 10/17-18
11	24-Oct	Ch.4 Sorting		
12	31-Oct	Ch.5 Trees	Project #5	
13	7-Nov	Ch.5 Trees (cont)	Written Exercise #2	Holiday 11/11
14	14-Nov	Ch. 6 Graphs	Project #6	
15	21-Nov	Ch. 6 Graphs (cont)		Holiday 11/24-25
16	28-Nov		Final Exam assigned	Classes end 12/2
17	5-Dec		Final Exam due	

\* This schedule is subject to change for the optimum benefit of the class as a whole. Therefore it is important to stay alert, read your email and read the discussion boards regularly.

### IX. Teaching Strategies

The primary teaching strategies for this course will be Blackboard notes, textbook readings, online discussions, written exercises and programming projects.

### X. Bibliography

Alfred V. Aho, Jeffrey D. Ullman & John E. Hopcroft. *Data Structures and Algorithms*. Addison-Wesley, 1983.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest & Clifford Stein. *Introduction to Algorithms, 2<sup>nd</sup> Edition*. MIT Press, 2001.

Allen B. Downey. *Think Python: An Introduction to Software Design*. Free book available online as PDF file at: <http://www.greenteapress.com/thinkpython/>

Michael Goldwasser & David Letscher. *Object-Oriented Programming in Python*. Prentice Hall Publishing, 2007.

John Zelle. *Python Programming: An Introduction to Computer Science, 2<sup>nd</sup> Edition*. Franklin Beedle & Associates, 2010.