COURSE NUMBER: CSC230

COURSE TITLE: Elementary Data Structures and Algorithms

CREDITS: 3 hours

PREREQUISITES/COREQUISITES:
- A grade of C or better in CSC130 or transfer credit for CSC130

This course is planned for undergraduate students in the Computer Science major or for students earning a minor in computer science.

INSTRUCTOR INFORMATION:
Lydia K Fritz, Lecturer, 152 Petty, lkfritz@uncg.edu.
Office Hours: Mon/Tues/Wed/Thurs 11:00am – 12 noon; other hours available by appointment.

CATALOG DESCRIPTION:
Advanced syntax of high level language taught in CSC 130. Emphasis on modularization and abstraction, big-O analysis of algorithms, design and use of abstract data types with various implementations.

STUDENT LEARNING OUTCOMES & COURSE TOPICS:
Upon successful completion of this course, a student should be able to
- apply the object oriented programming concepts object composition, inheritance and polymorphism
- design abstract data types and interfaces
- implement sequential abstract data types using both array and linked data
- describe characteristics of elementary searching and sorting algorithms
- implement recursive solutions to common computing problems
- evaluate algorithm performance using asymptotic analysis

Course topics include
- object composition and inheritance
- design and implementation of elementary container data structures, including arrays, linked-lists, stacks and queues
- mechanics and efficiency of elementary searching and sorting algorithms, including String handling algorithms
- recursion
- analysis of algorithms covered in this course

COURSE DELIVERY:
This course will consist of 150 minutes of lecture per week. Class lectures will be interactive and students will be expected to participate in class discussion.

TEACHING PHILOSOPHY:
Simply regurgitating facts is not enough to illustrate an understanding of the material covered in this, or any, course. An individual who possesses a deep and thorough understanding of concepts can explain them to others, both verbally and in writing. Unless knowledge can be communicated to an audience, it is useless. The tests in this course will include written, essay-type questions. Questions will be evaluated not only on content, but on how well the content is expressed.
EVALUATION AND GRADING:
- 10% class participation/attendance/in-class exercises
- 30% programming assignments
- 15% each for two written tests
- 30% cumulative final exam

Grading Scale


BRING YOUR BOOK TO CLASS EVERY DAY.

TOPICAL OUTLINE:
- 2-D Arrays
- String handling algorithms
- Abstraction, the ADT and interfaces
- Object composition and Inheritance
- Polymorphism
- Linked Data
- Data structures including sets, bags, lists, queues, stacks
- Searching and Sorting algorithms
- Recursive algorithms

EXERCISES:
Text chapter exercises should be completed as the chapter is being read. These exercises are good practice for tests. If time permits, I will answer questions about the exercises in class; if you need additional help, see me in my office.

PROGRAMMING ASSIGNMENTS:
- Programming assignments will be coded outside of class. Students are encouraged to use the Java tutors available in the Petty 211 lab. Schedule details will be available on Canvas by the end of the second week of class.
- Programs that do not compile will not be evaluated. A score of 0 will be given if a program does not compile, REGARDLESS OF REASON.
- All programs will be submitted in Canvas. Programs will not be accepted via email. Canvas is sometimes offline for maintenance or other reasons. This is not an excuse for late work. Plan accordingly and do not wait until the last minute to submit your work.
- Good programming practices must be followed. The instructor reserves the right to deduct for any assignment that is not well-documented, of poor readability, or of otherwise unprofessional quality.
- Programs must include initial comments. A short video is provided illustrating this (Canvas).
- See “Missed/Late Work” policy, below.
MISSED/LATE WORK POLICY:
- Makeup tests are not given
- The final exam must be taken. If the final is missed, the student will be given an incomplete in the course.
- Students with planned absences, whether for university events, religious observance, or other reason, are expected to make arrangements with the instructor to turn in assignments or take exams before the scheduled date of the assignment or test.
- All programming assignments will count – there are no dropped grades!
- Late work is accepted at a cost of 5 points per 12 hour period.

ACADEMIC INTEGRITY POLICY:
Work turned in for grading should be entirely your own. Students are expected to adhere to the UNCG Academic Integrity Policy, discussed in the first class and linked from the syllabus. See http://studentconduct.uncg.edu. Each assignment must include a statement that the work is your own.

ATTENDANCE POLICY:
Attendance is taken daily and is used when calculating your participation grade for the course. The university allows for a limited number of excused absences for religious observances --- students who plan to take such an absence should notify the instructor at least two weeks in advance so that accommodations can be made (also see the missed/late work policy below).

ELECTRONIC DEVICES POLICY:
The use of personal electronic devices is prohibited in this course. If you are seen using an electronic device, you will be asked to leave the classroom. If you have an emergency that requires the use of your phone, please exit the classroom before responding to the call/text/etc...

FINAL EXAMINATION:
A cumulative final exam is required and will be given during the time specified on the University Registrar’s Office Exam Schedule. You MUST take the final exam. Students who do not take the exam will be given a grade of Incomplete in the course.
- MW class exam is Monday, December 7 @ 3:30pm
- TR class exam is Thursday, December 3 @ 8:00am

CLASS HANDOUTS:
Any handouts used in class will be available through the Course Documents link on Canvas.

ANNOUNCEMENTS:
If the need should arise, any announcements to the class will be made through the Announcements page on Canvas, so check it often.

EMERGENCY PREPAREDNESS:
Closure of university facilities and classrooms in response to some emergency does not mean that this class is halted. Students should check Canvas for announcements about how the class will proceed in the event of such an emergency.