

CS 302 Data Structures

Fall 2009 – Dr. George Bebis

Catalog Description: Data structures and algorithms fundamental to computer science; abstract data-type concepts; measures of program running time and time complexity; algorithm analysis and design techniques.

Prerequisites: C or above in CS202. If you do not meet the prerequisite requirements for this course, you should see me immediately. *Credit hours:* 3.0

Meets: TR 1:00-2:15 PM (SEM 347)

Instructor: Dr. George Bebis

Office: 235 SEM

Phone: 784-6463

E-mail: bebis@cse.unr.edu

Course Webpage: <http://www.cse.unr.edu/~bebis/CS302>

Office Hours: Tuesday/Thursday 4:00 pm - 5:30 pm and by appointment

TA: Yanbo Li (HW & Quizzes), Nate Silva (Prog Assignments)

E-mail: liy19@unr.nevada.edu, masterdnater@gmail.com

Office: SEM 255A

Office Hours: TBA

Required Text:

C++ Plus Data Structures by N. Dale, Jones and Bartlett Publishers, 4th edition, 2007.

Optional Texts:

Practical Debugging in C++ by A. Ford and T. Teorey, Prentice Hall 2002.

Data Structures with C++ by W. Ford and W. Topp, Prentice Hall, 1996.

Data Structures and Program Design in C++ by R. Kruse and A. Ryba, Prentice Hall, 1999.

Data Structures and Other Objects Using C++ by M. Main and W. Savitch, Addison-Wesley, 1997.

Objectives

The purpose of this course is to introduce you to the exciting world of data structures, an issue central to the art of computer programming. At the end of the course you will be equipped with the tools of data organization to enable you to write simple, clear, and efficient programs. The course will be structured around a comprehensive set of computer assignments to enable you to get hands on experience. Our programming language of choice will be C++.

Course Outline (tentative)

- Software Engineering Principles (Chapter 1 - reading assignment)
- C++ Review
- Comparison of Algorithms
- Stacks & Templates
- Queues
- Unsorted and Sorted Lists
- Linked Structures
- Lists Plus
- Recursion
- Inheritance
- Binary Search Tree
- Heaps and Priority Queues
- Graphs
- Sorting and Searching (if time permits)

Exams and Assignments

Grading will be based on two exams, 6-8 quizzes, 6-8 homework assignments, and 4-5 programming assignments. Details are provided below:

- Quizzes will be announced at least one class period in advance.
- Homework problems will be assigned and collected for grading on a regular basis. Homework solutions will be made available within a week of the due date for the assignment.
- There will be two exams: a midterm and a final. The material covered in the exams will be drawn from the lectures, the quizzes, and the homework.
- There will be several programming assignments which will be done in groups of two. The goal of the assignments is to provide you with hands-on experience with handling various kinds of data structures. In the lectures, you will be introduced to data structures at an *abstract* level and in the assignments you will write code to *implement* and *use* these data structures. Specific details and due dates will be announced in class.

Course Policies

- Lecture slides, assignments, and other useful information will be posted on the course web page.
- Regular attendance is highly recommended. If you miss a class, you are responsible for all material covered or assigned in class.
- You should carefully read the section on Academic Dishonesty found in the UNR Student Handbook (copies of this section are available from <http://www.unr.edu/stsv/acdispol.html>) Your continued enrollment in this course implies that you have read it, and that you subscribe to the principles stated therein.
- The programming assignments are to be completed in groups of **two**, however, both members of a team are expected to fully understand the structure of the code and the implemented algorithms. Discussion of the programming assignments is allowed and encouraged. However, each team is expected to do its own work. **Assignments which are too similar will receive a zero.**
- **No** late homework or programming assignments will be accepted unless there is an extreme emergency.
- No incomplete grades (INC) will be given in this course and a missed quiz/exam may be made up only if it was missed due to an extreme emergency.

Useful Tips

Since the material in this course is highly integrated, a limited understanding of one topic will have a serious effect on the understanding of subsequent topics. You should expect to spend many hours on this course outside the classroom. Do not expect to fully understand the material covered in this class if you do not spend many hours in front of your computer.

Don't get behind in the programming assignments. Probably the main reason for students doing poorly in this course is getting behind in the assignments and never recovering. Design and implement in a top-down, modular fashion. Get something working that has the skeleton structure of what you need and then add features to it. Each time you add a feature, test it and make sure everything is still working. It can be tough to debug big programs if all you know is that the output is wrong and you are not sure anyone module is working. In addition, partial credit will be given for a program which at least partially works while it is very difficult to give credit for a program which may have many features but is not doing anything correctly.

Disability Statement

Any student with a disability needing academic accommodations is requested to speak with me or contact the Disability Resource Center (Thompson Building, Suite 101), as soon as possible to arrange for appropriate accommodations.

Grading Scheme

Midterm: 20%

Final: 20%

Quizzes: 15%

Homework: 15%

Prog Assign: 30%

A 90 and above

B 80-89

C 70-79

D 60-69

F<59

Important dates

10/20/2009 – Midterm exam

11/26/2009 - Thanksgiving

12/9/2009 – Prep Day

12/10/2009 - Final exam (noon – 2:00pm)

Outcomes and Objectives

Program Outcomes	Course Outcomes	Assessment Methods/Metrics	CS Program Objectives Impacted	CIE Program Objectives Impacted
1	Students demonstrate that they understand how data structures work and how to compare different implementations of a data structures and analyze their performance.	Exams, homework and quizzes measure level of understanding. .	1, 2, 3	1, 2, 3
3	Students demonstrate that they understand how to use appropriate data structures to solve a given problem.	Graded homework and programming assignments.	1, 2, 3	1, 2, 3
4	Students acquire an understanding of team dynamics by working in groups on programming assignments.	Graded project reports. Evaluate comments written by students discussing their experiences working in groups.	4	4
7	Students improve their communication skills by working in groups on programming assignments and giving a demo to the teaching assistant.	Graded project reports. Feedback from teaching assistant.	4	4
10	Students will improve their knowledge of contemporary issues by working on certain programming assignments related to image processing and computer vision.	Graded project reports related to image processing and computer vision.	1, 2	1, 2

CS/CIE Outcomes

(1) an ability to apply knowledge of computing, mathematics, science, and engineering
(2) an ability to design and conduct experiments, as well as to analyze and interpret data
(3) an ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs, within realistic constraints specific to the field
(4) an ability to function effectively on multi-disciplinary teams
(5) an ability to analyze a problem, and identify, formulate and use the appropriate computing and engineering requirements for obtaining its solution
(6) an understanding of professional, ethical, legal, security and social issues and responsibilities
(7) an ability to communicate effectively with a range of audiences
(8) the broad education necessary to analyze the local and global impact of computing and engineering solutions on individuals, organizations, and society
(9) a recognition of the need for, and an ability to engage in continuing professional development and life-long learning
(10) a knowledge of contemporary issues
(11) an ability to use current techniques, skills, and tools necessary for computing and engineering practice
(12) an ability to apply mathematical foundations, algorithmic principles, and computer science and engineering theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices
(13) an ability to apply design and development principles in the construction of software systems or computer systems of varying complexity

Computer Science Program Objectives:

Our graduates will have achieved:

1. a broad general education assuring an adequate foundation in science and mathematics relevant to computing.
2. a solid understanding of concepts fundamental to the discipline of computer science.
3. good analytic, design, and implementation skills required to formulate and solve computing problems.
4. the ability to function, communicate, and continue to learn effectively as ethically and socially responsible computer science professionals.

Computer and Information Engineering Program Objectives:

Within 3 to 5 years of graduation our graduates will:

1. be employed as computer engineering professionals beyond entry level positions or be making satisfactory progress in graduate programs.
2. have peer-recognized expertise together with the ability to articulate that expertise as computer engineering professionals.
3. apply good analytic, design, and implementation skills required to formulate and solve computer engineering problems.
4. demonstrate that they can function, communicate, collaborate and continue to learn effectively as ethically and socially responsible computer engineering professionals.