CSE 331: Data Structures

Fall 2001 Syllabus

Catalog Description:
CSE 331. Data Structures. Cr. 3. Fundamental techniques in the design and analysis of non numerical algorithms and their data structures. Elementary data structures such as lists, stacks, queues; more advanced ones such as priority queues and search trees. Design techniques such as divide-and-conquer. Sorting and searching algorithms. File compression, geometric methods, graph algorithms, dynamic programming, and string processing.
Prerequisites: CSE210, CSE232.

TEXTBOOK:

REFERENCE:

Instructor:
Prof. Jesús A. Izaguirre
E-mail: izaguirr@cse.nd.edu
Office: 326C Cushing Hall
Office Hours: M 2:00-5:00 PM, W 3:00-5:00 PM, F 2:00-5:00 PM.
Phone: (574) 631-7454

Teaching Assistants:
Scott Hampton
E-mail: shampton@cse.nd.edu
Office: 355N Fitzpatrick
Office Hours: Tu 2:00-4:00 PM, Th 4:00-6:00 PM
Phone: (574) 631-5772
Ping Yan

E-mail: pyan@cse.nd.edu
Office: 214 Cushing
Office Hours: Tu 4:00-6:00 PM, Th 2:00-4:00 PM
Phone: (219) 631-8720

Undergraduate Teaching Assistants:

Trevor Cickovski:
  E-mail: tcickovs@nd.edu

Branden Moore:
  E-mail: bmoore@nd.edu

Patrick Virtue:
  E-mail: pvirtue@nd.edu

Goals:

Enable students to choose appropriate data structures for given problems; review the important
data structures and some applications for them; give analytical reasons for the important data
structures; introduce a few algorithmic techniques, such as divide and conquer.

Computer Usage:

The programming assignments and some homework problems involve programming in C++ for
both UNIX and Microsoft’s platforms.

Laboratory:

None

Assignments:

- The student is required to submit an electronic half a page to a page text summary of the reading assigned
  before each class. Questions will be posted the afternoon of the previous class to guide the writing of the
  summary. You will be used to use these summaries and no others during the final exam. The summary
  should be named XXYY where XX is the month and YY the day of the due date and be in the summaries
directory in the dropbox directory hierarchy:

/afs/nd.edu/coursefa.01/cse/cse331.01/dropbox/student/summaries
/afs/nd.edu/coursefa.01/cse/cse331.01/dropbox/student/hw1
/afs/nd.edu/coursefa.01/cse/cse331.01/dropbox/student/hw2
/afs/nd.edu/coursefa.01/cse/cse331.01/dropbox/student/hw3
/afs/nd.edu/coursefa.01/cse/cse331.01/dropbox/student/hw4
/afs/nd.edu/coursefa.01/cse/cse331.01/dropbox/student/hw5
/afs/nd.edu/coursefa.01/cse/cse331.01/dropbox/student/hw6
/afs/nd.edu/coursefa.01/cse/cse331.01/dropbox/student/hw7
/afs/nd.edu/coursefa.01/cse/cse331.01/dropbox/student/hw8
/afs/nd.edu/coursefa.01/cse/cse331.01/dropbox/student/hw9
Throughout the semester **nine weekly assignments will be given.** These will range from mild to very complicated and are made to provide the student with the opportunity to improve his/her programming skills and to apply the concepts discussed in class. It is recommended that these assignments be started as soon as possible once assigned. These programs are due at the beginning of the class period on the due date. You are required to turn in an electronic copy of your source code to your dropbox. **No project will be accepted later than the deadline.**

**Help and review sessions:**

There will be review and help sessions every week at a time to be determined during class, attended by the TAs and some times by the instructor. You are encouraged to participate in them as much as possible.

**Honor Code:**

The standard Notre Dame honor policy is in effect. Particulars include:

**Students are encouraged to work in teams on each homework assignment** for **conceptual discussion** or **program planning**. However, at the time of carrying out the solution to a problem or implementing a program each person must do his/her own work. If in doubt, please consult the instructor or TAs. Use of the Internet as a **reference** is allowed but directly copying code or other information, unless specified otherwise and proper citation given, is cheating. Students are advised that it is cheating to copy, or to allow another person to copy, all or part of an exam or a project. Faking program output is also considered cheating.

**Grading:**

The course will be graded on the basis of homework assignments from chapter problems and other sources, the programming projects, two examinations, and a mildly cumulative final exam. The following weights will be used to compute the final grade:

- 20% - Reading summaries and at least 85% attendance
- 20% - Weekly assignments
- 20% - 2 take home exams
- 20% - Final project, paper, and presentation
- 20% - Final exam

**Final Examination:**

The final exam will be held during the University scheduled time. For our term, this is **Dec. 19, 8 - 10 AM.** Students should determine their final exam schedule for all of their courses and contact the Instructor if a problem exists as soon as possible.

**Discrepancies:**

If you have any questions regarding how any assignment or test is graded and you think that you deserve more points than you received, you must see the TA or Instructor within one week of the time the assignment is first returned to the class. No claims, justifiable or not, will be considered after this deadline.

**Course Web page:**
The course has extensive information posted on the web. Lecture slides, programming project and homework assignments and solutions, exam solutions and other useful information will be posted. Be sure to check it often for up to date information on the course. Questions on this site should be directed to the class TAs.

**Computer Use:**

All students should have an account on the engineering computer cluster. All the tools (e.g. emacs, CC, etc.) that you will need for the programming problems in this course will be available on the cluster. There will be directories established for electronic submission of programming assignments. The engineering cluster is the only computing environment guaranteed to be supported for this course. The Sun SPARC architecture is the only computer architecture guaranteed for this course. Students who desire to work in other environments or architectures may do so, but must also assume full responsibility for porting any necessary software, data, etc. to the other environment/architecture.

**KEYS TO SUCCESS**

To obtain a satisfactory grade in this course, the following are some time tested techniques which should help.

- Come to class and be on time. Believe it or not, even if you are not at your best, just being in class always helps.
- Complete the readings prior to lecture.
- Keep up with the work. Do not fall behind and think that you will be able to catch up later in the course. Late penalties are severe and work often builds on previous work.
- Participate in review and help sessions and come to office hours with the instructor and the TAs. They are abundant and can be a source of more personalized instruction.
- Notify the Instructor **immediately** if something happens which causes you to miss class for any length of time. Informing the Instructor a week or more after the fact will not be received well and penalties accrue until the Instructor is notified.
- Remember that if you miss class, it is YOUR responsibility to find out what you missed and to get caught up with the work. Therefore, if you can, contact a classmate as soon as possible to learn what was covered that day and to obtain notes, handouts, announcements, etc.
- Missing a class is NO excuse for late work. You MUST get the assignment turned in when it is due. This might require you to turn it in early, send it with a classmate, or even fax it or e-mail it in.

**Recommendations for successful completion of the Programming Projects:**

- Think and Plan before you start programming.
- Start early!
- Do not be reluctant to ask for help. Do not wait to ask for help if you are feeling lost.