Syllabus: CSCI-C 243
Introduction to Data Structures

Indiana University East

Fall, 2012

Version 1.5, 2012 Aug 24. OFFICIAL.
Course Web Site: http://mypage.iu.edu/~gdweber/csci/c243/

Scheduling Information

Time and location: MW 7:00–8:15 p.m., TR 102
Frequency: Every fall semester
Start date: Mon, Aug 30
Last date to register: Sep 1
Final exam: Wed, Dec 17

Introduction

Course Description

C243 Introduction to Data Structures (3 cr.). Prerequisite: C202. Introduction to data structure concepts and common applications. Structures to be discussed include strings, lists, queues, stacks, graphs, trees, sequential files, random files, and indexed sequential files. Practical applications and algorithms are stressed.

Prerequisite

Students entering this course should have had two semesters of prior programming experience, preferably including at least one semester of Java. Students whose prior experience is with C++, C# (C-sharp), or similar languages should be able to adapt to Java with some extra work. At IU East, the course normally taken to satisfy the prerequisite is INFO-I211, Information Infrastructure II. Familiarity with algebra, including polynomial, exponential, and logarithmic functions is expected.
Comments

This course covers tools for managing complexity in software development, measuring performance, and demonstrating correctness of programs. Data structures covered include lists, stacks, queues, trees, balanced search trees, priority queues, hash tables, and graphs. We will not cover file organizations (i.e., sequential, random, and indexed sequential).

This course should be illuminating and useful to anyone who is interested in how to organize data in computer memory for effective information processing. It is a required course for the concentration in Management Information Systems (MIS) of the Bachelor of Science in Business, and a good elective for the B.S. in Informatics.

Learning Objectives

Campus Learning Objectives

These course learning objectives relate to the campus undergraduate learning objectives\(^1\) as follows:

“2. [Depth] Educated persons should have achieved depth in some field of knowledge. . . .”

This course provides in-depth knowledge of data structures for majors in computer science, informatics, and MIS, and other interested students.

“3. [Expression] Educated persons should be able to express themselves clearly, completely, and accurately. . . .”

We seek to develop good programming style, including systematic documentation practices, to express the ideas of a program to ourselves and to other programmers. In particular, we will be reading and writing documentation of abstract data types.

“4. [Computation] Educated persons should be able to relate computational skills to all fields so that they are able to think with numbers. . . .”

Through learning how to analyze the running time of algorithms, we develop our computational ability.

“5. [Problem Solving and Critical Thinking] Educated persons should have the ability to develop informed opinions, to comprehend, formulate, and critically evaluate ideas, and to identify problems and find solutions to those problems. . . .”

Throughout this course, we consider organized approaches to the solution of complex problems. As programmers, we seek to enhance our skills in designing, coding, and debugging more complex programs. Critical thinking applied to algorithms and programs includes reasoning about their correctness and efficiency, ideally leading to a proof, and systematic testing. As we design, code, debug, and test programs, we will be exercising and developing our critical thinking and problem solving skills.

Course Learning Objectives

- **Data structures** are ways of organizing data in computer memory (and secondary storage) so that it can be processed efficiently. Abstract data types (ADTs) are programmer interfaces to data structures. Through this course, students should develop the ability to make intelligent choices among abstract data types and the data structures that underly them.

- Another important skill is the use of abstractions to manage program complexity so that we can develop larger, more complicated programs. We document these abstractions so that programmers (others or ourselves) will know how to use them.

- We also seek to polish our understanding of recursion as a problem solving technique.

Learning Resources


**Oncourse** https://oncourse.iu.edu/
(Please memorize or set a bookmark!)

**Course Web Site** http://mypage.iu.edu/~gdweber/csci/c243/
(Please memorize or set a bookmark!)

**Software** Primarily, we will be using Emacs or another text editor to prepare our programs, and a Java compiler and interpreter.

Most of our programming will be in Java. Java 2 SDK (or JDK) version 1.7 (marketing-speak: “7.0”) is recommended; version (“5.0”) or later is acceptable. Sun Java or OpenJDK will work fine for most applications. Our programs depend on features introduced in Java 1.5; earlier versions will not work. Also, you need the Java compiler, not just the interpreter and run-time; the compiler is included in downloads called “SDK” or “JDK” (Software or Java Development Kit).

We will also try a little Haskell programming for functional data structures. For this we’ll use the Glasgow Haskell Compiler, GHC.

I will encourage you to use a version control system, such as darcs or git.

Students will need a web browser that adequately implements the standards (“recommendations”) of the World-Wide Web Consortium (W3C). The most recent versions of popular browsers—Mozilla Firefox, Chromium or Google Chrome, Opera, Apple Safari, and Microsoft Internet Explorer—are doing a mostly commendable job of trying to keep up with the standards (which keep changing). If you have an older web browser (especially Internet Explorer < 9), you should upgrade it to the latest stable version. No matter which “brand” of browser you use, if it’s below the latest stable version, you are open to

---

2 http://www.w3c.org/
security hazards as well as not having the latest and greatest browsing experience, so upgrade today!

Other software which some students may find useful includes a PDF reader, a secure shell client, VNC, a good operating system (such as Linux or Unix—including Mac OS X, can be a virtual machine).

All software required or recommended for this course is available for student use on our Linux server. Students wishing to install the software on their own computers should consult the Courseware Links page\(^3\) and/or INFO-I 210 Computer Setup Guide\(^4\) for details.

Secure shell (ssh) enables students to login remotely to the Linux server, but in “text-only” mode, i.e., without graphics. Using VNC in combination with ssh enables remote login with the full graphical power of the X Window System (X11); this requires a reasonably high speed internet connection. X11 can also be used remotely without VNC, but that needs an even faster network connection.

Learning Activities and Measurement of Learning

We learn by reading, listening, and discussing. We learn and demonstrate our learning by writing and testing programs. We also demonstrate our learning by quizzes and exams.

Reading

Every student is expected to read ahead, so that you have read the assigned chapters of the textbook, and other assigned readings, before they are presented or discussed in class. The instructor may post notes on some or all of the readings, but reading the notes is no substitute for the readings themselves or for taking one’s own notes.

In-Class Discussion

Every student should attend class regularly, and having read the assigned chapters in advance, participate in discussion. Either ask about something you did not understand, or be prepared to answer about something you did understand.

E-Discussion

Electronic channels for discussion supplement the regular in-classroom discussion. These include the Oncourse discussion forum, which is recommended, except use Oncourse Messages if privacy is needed.

\(^3\)http://mypage.iu.edu/~gdueber/software/courseware.html
\(^4\)http://mypage.iu.edu/~gdueber/info/i210/dest/00-0-setup-guide.html
Labs

The labs, or programming projects, for this course are intended to be larger and more complex than the labs in INFO-I210 and I211. Students should plan their time accordingly.

There will be about six labs.

Quizzes

We will have seven quizzes (about one every two weeks). The best 5 of 7 quizzes will count towards the student’s grade; that is, we will drop the lowest 2 of 7 quiz scores.

All quizzes are open-book, open-notes, and on the honor code, with no human help allowed. Quizzes will be given to take home and will be due the following class day. Late quizzes will not normally be accepted.

Examinations

The midterm exam will cover chapters 1–9. The final exam will cover the entire course, but will emphasize the last half of the course, chapters 10–12 and 14–17.

These exams will be closed book, closed notes, using pencil and paper, and not using an electronic computer. Exams are given in the classroom, except students in the online section will take their exams through the IU East Testing Center or with a proctor approved by the instructor.

No late or makeup exams will be allowed without prior arrangement and documented, serious reasons. “Prior arrangement” means, if at all possible, contact the instructor before the exam is given. In case of emergencies, the instructor should be notified at least by the end of the day.

Activities Summary: Determination of Course Grade

Table 1 shows an estimate of the number of pieces of graded work of each type and how they contribute to the course grade. Table 2 shows the grading scale.

<table>
<thead>
<tr>
<th>Component</th>
<th>Points</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 labs, 50–64 points each</td>
<td>344</td>
<td>50</td>
</tr>
<tr>
<td>Quizzes: best 5 of 7, 20 points each</td>
<td>100</td>
<td>14</td>
</tr>
<tr>
<td>Midterm exam</td>
<td>100</td>
<td>14</td>
</tr>
<tr>
<td>Final exam</td>
<td>150</td>
<td>21</td>
</tr>
<tr>
<td>TOTAL</td>
<td>694</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 1: Estimated Point Distribution
<table>
<thead>
<tr>
<th>Percent Grade</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>96.667–100.000</td>
<td>A+</td>
</tr>
<tr>
<td>86.667–89.999</td>
<td>B+</td>
</tr>
<tr>
<td>76.667–79.999</td>
<td>C+</td>
</tr>
<tr>
<td>66.667–69.999</td>
<td>D+</td>
</tr>
<tr>
<td>93.333–96.666</td>
<td>A</td>
</tr>
<tr>
<td>83.333–86.666</td>
<td>B</td>
</tr>
<tr>
<td>73.333–76.666</td>
<td>C</td>
</tr>
<tr>
<td>63.333–66.666</td>
<td>D</td>
</tr>
<tr>
<td>90.000–93.332</td>
<td>A−</td>
</tr>
<tr>
<td>80.000–83.332</td>
<td>B−</td>
</tr>
<tr>
<td>70.000–73.332</td>
<td>C−</td>
</tr>
<tr>
<td>60.000–63.332</td>
<td>D−</td>
</tr>
<tr>
<td>0.000–59.999</td>
<td>F</td>
</tr>
</tbody>
</table>

Table 2: Grading Scale

Advice

Read the book. The instructor will not read the book for you. Take notes as you read. Write down questions about anything you don’t understand, and bring them to class. Ask them there.

Do the assigned programming problems and as many additional problems as you can. Learning to program is a lot like learning to play the piano or play basketball: you can’t do it without practice, and the more practice, the better.

Come to class regularly and participate.

If you need help, ask the instructor. There are many channels for giving help. If the regular office and lab hours are inconvenient, we can arrange something else.

Lab Facilities

The lab is in Tom Raper Hall 102. Please check the schedule by the lab door to see when it is not reserved for a class. Starting in Fall, 2012, there are no Linux workstations in the lab; each computer in the lab (and I believe in all other campus labs) has a Windows operating system with PuTTY and VNC for connecting to our Linux server.

Students using a secure shell client may log in remotely to the Linux server, merlin.iue.edu. With VNC you can have a graphical remote login, but this works poorly with a slow Internet connection. **Note: merlin.iue.edu is not a web server, and you cannot “open it” in a web browser. You must use software such as secure shell, PuTTY, or VNC to access the merlin server!**

Please contact the IT Help Desk (Hayes Hall 169, telephone 765-973-8375) for any problems of lab access including locked doors, login and password. The lab should be open (unlocked) on the same schedule as other campus computer labs: Monday–Thursday 8 a.m.–9 p.m. and Friday 8 a.m.–5 p.m.

Please see the Linux Lab Manual⁵ for further information.

⁵http://mypage.iu.edu/~gdueber/doc/labman/xhtml/index.html
Notices

Late Homework

One characteristic of good software is that it is delivered “on time,” and the same is true of homework, such as labs. Work that is turned in on time may earn full credit. Work turned in late will earn partial credit, up to 90% for one week late, up to 80% for two weeks late. Unless a student is entitled to a grade of “Incomplete,” no assignment will be accepted after the last day of class before the final exam.

I do not normally accept assignments that are more than two weeks late; however, in case of extraordinary circumstances, discuss the situation with me (earlier is better) and we may work something out.

Program development is hard to schedule precisely, even for experienced programmers. The instructor will make reasonable efforts to accommodate variations in learning rate, and allow revision and resubmission of seriously flawed programs. Students are strongly cautioned against falling behind in programming assignments, as they help to prepare for exams.

Attendance

IU requires students to attend class.

Merely not attending does not withdraw a student from a course; see “Withdrawal” below.

Students who fail the course due to non-attendance will receive a grade of FN or FNN; these grades can result in termination and reversal (paying back) of financial aid.

Withdrawal

Withdrawing from a course requires a withdrawal form. Normally, withdrawal must take place on or before the “last date for withdrawal with an automatic W”, which this semester is (Oct 26). Withdrawal after that date requires the instructor to determine the grade as W, meaning the student was passing at the time of withdrawal, or WF, meaning the student was failing.

Helping, Cheating, and Academic Honesty

See the IU East Standards of Student Conduct Policy\(^6\) and the Indiana University Code of Student Rights, Responsibilities and Conduct,\(^7\) especially the section “Academic Misconduct.”\(^8\)

Strict academic honesty is expected of all students. The IU code of student conduct provides serious penalties for cheating. All work turned in for credit must be substantially the work of

\(^6\)http://www.iue.edu/organizations/senate/documents/Policies/studentaffairs/StudentConductPolicy090407.pdf
\(^7\)http://www.iu.edu/~code/
\(^8\)http://www.iu.edu/~code/code/responsibilities/academic/index.shtml
the student (or students, if teamwork is authorized) turning it in. Other students needing help, except for simple questions, should be referred to the instructor.

It is part of my responsibility as instructor to help students who are having difficulty with their assignments. Don’t be ashamed or embarrassed to ask me for help! I want to help! (By the way: usually, I will try to help a student to think through the solution rather than directly provide an answer.)

Copying another person’s work is cheating, and so is providing the original work to another student for copying. In such cases, both students are equally guilty and will be equally punished. Do not share your work with other students or leave it lying around for anyone to pick up. There are usually many different ways to solve a problem; therefore, identical or very similar solutions are prima facie evidence of cheating.

Restrictions on Copying and Distribution of Class Materials

All of the materials that I post that are my original material, including any of my own lecture notes that I post for your reference, are materials that you may use freely for the purposes of your study within this course and other academic courses. You may not copy or distribute them, electronically or otherwise, for any other purpose without asking me first. Similarly, photographing and audio or video recording of our classes (and posting said pictures/files to YouTube or another site) is not permitted, except as authorized for the purpose of accommodating a documented disability.

Disabilities

Students with disabilities (including physical, mental, sensory, and learning impairment) and wishing to receive auxiliary aids and services (“accommodations”) should fill out a Disability Accommodations Request Form and send it to the Student Support Services Office, early in the semester, and inform their instructor if appropriate. See the Student Support Services site9 or contact Student Support Services for additional information.

Religious Observances

Students needing an accommodation for a religious observance (for example, rescheduling an exam from a date on which work is prohibited) should communicate their needs to the instructor, in accordance with the IU East “Religious Observances Policy”10.

Student Athletes

Student athletes who are unable to attend class(es) because of any IU East athletic events must inform the instructor, during the first week of class, about the conflicting dates. The instructor

9http://www.iue.edu/support/
will inform students if accommodations can be made, the nature of the accommodations, and the accommodations will be written and signed by instructor and student, with a copy available for the coach. For details, see the IU East “Policy for the Approved Absence of Students Participating in Athletic Events”.

**Contacting the Instructor**

**Name:** Gregory D. Weber  
**Office location:** HY 255T  
**Office hours:** See [http://mypage.iu.edu/~gdweber/contact/schedule.html](http://mypage.iu.edu/~gdweber/contact/schedule.html) for my (possibly updated) normal schedule.  
Additional hours by appointment.  
**Telephone:** (765) 973-8420 (voice); (765) 973-8550 (FAX).  
**Messages:** Please use Oncourse Messages instead of email for class-related communications. For other purposes, such as advising, see contact information at [http://mypage.iu.edu/~gdweber/contact/info.html](http://mypage.iu.edu/~gdweber/contact/info.html).  
**Other channels:** (by appointment only)  
IU’s “Connect” Adobe Breeze service  
Jabber/XMPP chat: magister.informaticae@jabber.org  
SIP phone: magister.informaticae@ekiga.net  
**Personal home page:** [http://mypage.iu.edu/~gdweber/](http://mypage.iu.edu/~gdweber/) (this is *not* the course web site!)

---

Schedule

As of 2012 Aug 24. Subject to change. A and B indicate the pages to be read before the first and second day of a unit, respectively, including the first day of class. The precise dates of quizzes and labs will be announced.

<table>
<thead>
<tr>
<th>Dates</th>
<th>Topics and Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aug 27</td>
<td>Overview; Appendix A, Java review.</td>
</tr>
<tr>
<td>Aug 29</td>
<td>Chapter 1. Encapsulation. Lab 1 out (encapsulation).</td>
</tr>
<tr>
<td>(Sep 3)</td>
<td>Labor Day: No class.</td>
</tr>
<tr>
<td>Sep 5</td>
<td>Ch 2. Polymorphism.</td>
</tr>
<tr>
<td>Sep 12, 17</td>
<td>Ch 4 (A: 87–100; B: 100–110). Stacks and queues. Lab 1 in; 2 out (stacks).</td>
</tr>
<tr>
<td>Sep 26, Oct 1</td>
<td>Ch 6 (A: 157–168; B: 168–178). Linked structures. Lab 2 in; 3 out (linked lists).</td>
</tr>
<tr>
<td>Oct 8</td>
<td>Ch 8. Searching and sorting.</td>
</tr>
<tr>
<td>Oct 15</td>
<td>Review. Lab 3 in.</td>
</tr>
<tr>
<td>Oct 17</td>
<td><strong>Midterm exam</strong> (online: Oct 16–18), chs 1–8. Lab 4 out (quicksort).</td>
</tr>
<tr>
<td>(Oct 26)</td>
<td>Last date to withdraw with an automatic W.</td>
</tr>
<tr>
<td>Nov 14</td>
<td>Ch 16. Memory management. Lab 5 in, 6 out (graphs).</td>
</tr>
<tr>
<td>(Nov 19–24)</td>
<td>Fall break; Thanksgiving; no class.</td>
</tr>
<tr>
<td>Dec 10</td>
<td>Review, emphasizing chs 9–11, 14–17.</td>
</tr>
<tr>
<td><strong>Dec 17</strong></td>
<td><strong>Final exam</strong> (online: Dec 13–17), Chs 1–11, 14–17.</td>
</tr>
</tbody>
</table>