Computer Science 228
Introduction to Data Structures
Spring 2015

Note: Remember to refresh your browser to see updates to this document. Rooms, office hours, and other details are subject to change!

Last modified on January 8, 2015

Instructors

David Fernández-Baca (Section A)
Office: 111 Atanasoff Hall
Phone: 294-2168
E-mail: `fernande@iastate.edu`
Lecture: MWF 11:00am–11:50am, in Ross H 0124
Office Hours: TBD

Yan-Bin Jia (Section B)
Office: 207 Atanasoff Hall
Phone: 294-2577
E-mail: `jia@iastate.edu`
Lecture: MWF 12:10pm–1:00pm, in Mol-Bio 1414
Office Hours: W 3:00pm–4:00pm

Instructors' office hours will be held in their respective offices.

Teaching Assistants

		Office Hours
Arko Basu	abasu@iastate.edu	TBD
Caleb Brose	cmbrose@iastate.edu	TBD
Yuxiang Chen	yuxiangc@iastate.edu	TBD
Christopher Fogerty	cfogerty@iastate.edu	TBD
Austin Liu	azliu@iastate.edu	TBD
Monica Kozbial	mkozbial@iastate.edu	TBD
Alex Maxwell	maxwella@iastate.edu	TBD
Lingjian Meng	lmeng@iastate.edu	TBD
Bryan Passini	bpassini@iastate.edu	TBD
Blake Skaja	bskaja@iastate.edu	TBD
Andrew Snyder	acsnyder@iastate.edu	TBD
Jacob Stimes	jstimes@iastate.edu	TBD
Caleb Van Dyke	cvandyke@iastate.edu	TBD
Anthony Wilson	apwilson@iastate.edu	TBD
Chen-Yeou Yu	cyy232@iastate.edu	TBD
Ryan Young	rtyoung9@iastate.edu	TBD

TA office hours will be held in

- *Pearson 113*, between 12:10pm and 6:00pm, and
- *Pearson 145*, for all other times.

Recitation Sessions

Section	Day & Time	TAs	Place
1	R 10:00am-10:50am	TBD	Pearson 1105
2	R 2:10am–3:00pm	TBD	Gilman 2109
3	R 1:10pm–2:00pm	TBD	Gilman 2109
4	R 4:10pm–5:00pm	TBD	Sweeney 1126
5	R 3:10pm–4:00pm	TBD	Gilman 2109
6	T 9:00am–9:50am	TBD	Gilman 1810
7	T 2:10pm–3:00pm	TBD	Gilman 2109

- There will be no recitation in the first week of class (Jan 12–16)
- The first recitations are on Tuesday January 20.

Blackboard Learn

We will be using Blackboard Learn for this course. In particular,

- announcements about the course, such as assignments, due dates, and exam locations, will be made via Blackboard Learn,

- clarifications and answers to common questions about homework will be announced through the Blackboard Learn discussion boards, and
- unless announced otherwise, all programming assignments must be submitted electronically via Blackboard Learn.

We expect you to check Blackboard Learn regularly for announcements and homework/exam information.It is your responsibility to know how to log in, read announcements, use the discussion boards and submit homework. You can log into Blackboard Learn with your ISU NetID and password at

`https://bb.its.iastate.edu/`

Choose Com S 228 after you log on.

General Contact Instructions

If you have a general question about the course, about an assignment, or about Java, the best place to start is on one of the Blackboard Learn discussion topics, where it will be seen by the instructor, the TAs, and the rest of the class. However, please do not post personal information, or source code for an assignment, on Blackboard Learn.

If you need to send an email to the instructors or the TAs, please begin the subject line with "CS 228".

Course Description

The formal catalog description is as follows.

> **COM S 228. Introduction to Data Structures.**
> (3-1) Cr. 3. F.S. Prereq: Minimum of C- in 227, credit or enrollment in MATH 165 An object-oriented approach to data structures and algorithms. Object-oriented analysis, design, and programming, with emphasis on data abstraction, inheritance and subtype polymorphism. Abstract data type specification and correctness. Collections and associated algorithms, such as stacks, queues, lists, trees. Searching and sorting algorithms. Graphs. Data on secondary storage. Analysis of algorithms. Emphasis on object-oriented design, writing and documenting medium-sized programs. This course is designed for majors

Com S 228 is a cornerstone of the Computer Science curriculum. The course introduces the fundamentals of *algorithms* and *data structures*. An algorithm is a step-by-step procedure for performing a task — e.g., sorting a list of numbers. A data structure is a method of organizing data — e.g., a dictionary — so that it can be accessed and updated efficiently. Although we use Java as the programming language and object-oriented programming as the paradigm, you will find that the usefulness of algorithms and data structures extends well beyond the specific language and paradigm used.

Topics

- Java and Object-Oriented Concepts
- Exception Handling

- Analysis of Algorithms
- Sorting Algorithms
- Generics in Java
- Collection Interface and Iterators
- Lists
- Stacks
- Queues and Priority Queues
- Trees, Binary Search Trees
- Graphs and basic Graph Algorithms
- Maps

Course Objectives

At the end of Computer Science 228, students are expected to

- be able to write and debug well-structured object-oriented programs of 2000 or more lines of Java code,
- be able to implement a Java class given a specification,
- be able to apply OO principles such as encapsulation and inheritance,
- be able to implement basic data structures in Java including expandable arrays, linked lists, trees, heaps, and hashtables,
- understand abstract data types (ADTs) including lists, stacks, queues, sets, and maps, be familiar with different ways of implementing these ADTs, and know common algorithms for using or manipulating these ADTs, and
- be able to perform runtime analysis of simple algorithms, and know several searching and sorting algorithms and their runtime analysis.

Course Outcomes

This course has three major ABET outcomes.

A. An ability to apply knowledge of computing and mathematics appropriate to the discipline.

B. An ability to analyze a problem and identify and define the computing requirements appropriate to its solution.

C. An ability to design, implement, and evaluate a computer-based system, process, component or program to meet desired needs.

These outcomes are broken down as follows.

A.1 Be able to perform runtime analysis of simple algorithms and sorting & searching algorithms.

A.2 Be able to form and check conditions for the valid values of variables at certain stages of a program.

B.1 Understand abstract data types including lists, stacks, queues, sets, maps etc.

B.2 Be able to specify the input, output, correctness, efficiency, and maintainability requirements of a programming solution.

C.1 Be able to understand sorting, searching and graph algorithms in order to design and implement efficient solutions.

C.2 Be able to implement abstract data types such as lists, stacks, queues, binary search trees, and priority queues.

C.3 Be able to determine appropriate data structures and algorithms appropriate for program requirements.

C.4 Be able to use object-oriented programming principles in problem solving.

C.5 Be able to write and debug well-structured, object-oriented and properly documented programs of up to 1000 lines of Java code.

Textbook

There is no required textbook for this course; instead, we will mainly rely on lecture notes. For additional course material, you may consult a book such as

> Simon Gray (2007). *Data Structures in Java: From Abstract Data Types to the Java Collections Framework.* Pearson/Addison Wesley.

Lecture Attendance

We do not take attendance. Students are expected to attend all lectures and recitations. If you do not want to attend classes, it is your responsibility to find out what was covered and to learn it on your own.

Exams

Exam 1	Thursday Feb 19	6:45pm–7:45pm
Exam 2	Thursday Mar 26	6:45pm–7:45pm
Final Exam	TBD	TBD

You must bring your university ID to all exams.

If you cannot attend an exam, you must notify your instructor ***at least one week prior to the exam*** to make other arrangements. The instructor will normally adhere to ISU policies regarding exam scheduling.

Midterm exams will be returned in recitation. If you feel that an error was made in the grading of an exam, please return it to your instructor with a brief statement ***in writing*** indicating where you think the error was made. Such appeals must be made within one week of the date on which the exams are returned in recitation. Note that a regrade may result in a lower score.

The final exam will not be returned to you, but you can make an appointment with the instructor to review your test.

If you have a conflict with a scheduled exam, you must inform your instructor at least one week prior to the exam to resolve the conflict. The instructor will follow ISU polices on exam conflict resolution.

Homework

We expect to give five programming projects. Assignments will be posted on Blackboard Learn in the Assignments section. You should expect the assignments to be somewhat longer and more complex than those from Com S 227. Due dates may overlap.

The Clarification Thread

The homework specification may also include design issues that require further clarification, and it is part of your job to identify such issues and resolve them (well in advance of the deadline). Clarifications that are believed to be relevant to all students will be posted on Blackboard Learn in the "official clarification" discussion thread. ***All clarifications that are posted more than 24 hours before the assignment deadline are considered part of the homework specification*** and you may lose points if you ignore them.

Compilation and Runtime Errors

All code submitted must compile under the Java 1.7 JDK compiler. ***If your code does not compile, you will normally receive zero points for the assignment.*** Similarly, you are responsible for fully testing your code, whether or not any sample test code is posted for you to try and whether or not the test code is part of the assignment submission. You will lose additional points for runtime errors (e.g., infinite loops, uncaught exceptions, memory leaks, etc.) that occur during testing.

Documentation and Style

Documentation and style will count for 10 to 20 percent of each assignment. In particular, here are a few things to note:

- Each class and method must have a complete and correctly formatted javadoc comment.
- Methods should generally not be more than 60 lines long; complex tasks should be factored into private helper methods.
- Duplicated code is generally a bad idea.
- Internal (//-style) comments should be included appropriately. (Remember that a comment should precede the code it describes and should be indented to the same level.)

- Choose an indentation and formatting style and use it consistently. (FYI, Eclipse makes this very easy for you: Ctrl-Shift-F)

Submissions and Feedback

Programming assignments may be turned in **up to 24 hours late** with a 25% grading penalty. **You will not be able to turn in an assignment 24 hours after its due date**. Weekends and university holidays are counted in the 24 hours. Results will be returned via Blackboard Learn. If you feel an error has been made in grading your assignment, you must make an appeal to the TA that graded it within one week of the date when the results were made available in Blackboard Learn. The appeal should be made during the TA's office hours, or by email if you cannot attend the TA's office hours. Note that a regrade may result in a lower score. If you are not satisfied with the response from the TA, please contact the instructor promptly.

Correct submission of an assignment is your responsibility. Remember that when submitting an assignment via Blackboard Learn, you can immediately check whether the submission was successful, and you can always download your submission and verify that it is what you intended. Don't forget that with Blackboard Learn, it is **not** enough to just upload your work; ***you must also to click the "submit" button***. Detailed instructions will be provided prior to the first due date.

Grading

Your grade will be based on your performance in the three exams and five assignments and your participation in activities during class time with the following weights.

Assignments (5)	40%
Exam 1	17%
Exam 2	17%
Final Exam	26%

The next table gives a rough idea of the letter grade you will receive depending on your total score.

at least 88	A
at least 83 but less than 88	A-
at least 78 but less than 83	B+
at least 70 but less than 78	B
at least 65 but less than 70	B-
at least 60 but less than 65	C+
at least 55 but less than 60	C
at least 50 but less than 55	C-
at least 45 but less than 50	D+
at least 42 but less than 45	D
at least 39 but less than 42	D-
less than 39	F

Academic Dishonesty Policy

Unless specifically instructed otherwise, every assignment and lab activity is to be the product of your own intellectual effort and is to be done on your own.

Any violation of this rule will be considered academic dishonesty, otherwise known as cheating. Anyone found responsible of academic dishonesty will receive an automatic F in this course. Additionally, we will adhere to university policies regarding academic dishonesty, which means that you may receive any of the penalties described in the Policies and Practices section of the Student Handbook under Academic Dishonesty:

`http://www.public.iastate.edu/~catalog/2005-07/geninfo/dishonesty.html`

There are two forms of academic dishonesty to be careful about.

- *Plagiarism:* If you present the work of someone else as if it were your own, you are guilty of plagiarism. Plagiarism is unethical at every level of every profession.
- *Tendering information:* In a university environment (such as in this course), if you make your assignment available for someone else to copy, you are guilty of tendering information, which is also considered to be a form of academic dishonesty.

Here are just a few examples of things you ***may not do*** when working on an assignment:

- Look at another students assignment source code.
- Let another student look at your assignment source code.
- Write any part of assignment source code with another student.
- Type in a solution with another student and each turn in a copy.
- Write a program with another student on paper and type it up separately.
- Divide the work so that you and another student each write part of a program (for example, three methods each), then combine them, and each turn in a copy.
- Discuss the code of assignments in detail with another student, but type it in separately.
- Copy source code from an outside source (such as a web site) and submit it as your work.
- Modify code from an outside source (such as a web site) and submit it as your work.
- Give another student your password.
- Share any part or all your code for assignments to others by any means, whether by paper, email, ftp, Blackboard Learn discussion, shared network folder, storing on a public machine, letting someone else look at your screen, reading code out loud while someone takes notes, posting in an online forum, etc.

There is one exception to the above rules: code may be shared after it can no longer be turned in for credit (i.e., after the late deadline — normally 24 hours after the deadline, counting weekends and holidays).

Here are some things that you ***may*** do when working on an assignment.

- Talk about, and write down ideas about, how to do an assignment, as long as you do not write actual code or pseudocode together.
- Share test code or test data files that will not be turned in.

- Share and discuss code that was presented as an example in class.
- Share and discuss code for programming assignments ***after*** the late deadline for the assignment has passed (the late deadline is normally 24 hours after the deadline, counting weekends and holidays).

The rules against sharing code are not intended to prevent students from studying and working together. Remember that all the ideas and techniques you need to do an assignment will have been presented as examples in class or in the book, and you can discuss all such examples freely. In general, it is acceptable to discuss how to do the homework with other students, but when it is time to sit down and write your code, ***you must be able to produce the entire result without help from anyone else***.

If you get help from any source besides the lecture notes, instructors, or TAs, you ***must cite these sources***. For example, if you look at code on a website to get ideas for how to do the homework, you must clearly indicate what the website was and what information you got from it. You may not receive full credit for that work, since you aren't handing in something that is solely your own work, but at least you will not be guilty of plagiarism.

A W-W-W-Warning

Just as a word to the wise, note that random examples you find on the web are likely to mislead you more than help you, since those who post them have no idea of the ideas and techniques being covered in our particular class. You will be much better off reviewing examples from class and posting your questions on the Blackboard Learn discussions, where the instructor, TAs, and other students going through the same experience can help you.

Disabilities

Iowa State University complies with the Americans with Disabilities Act and Section 504 of the Rehabilitation Act. Any student who may require an accommodation under such provisions should contact the instructor as soon as possible and no later than the end of the first week of class or as soon as you become aware. Please obtain a SAAR form verifying your disability and specifying the accommodation you will need. No retrospective accommodations will be required in this class.

Dead Week Policy

This class follows the Iowa State University Dead Week policy as noted in Section 10.6.4 of the Faculty Handbook.

Harassment and Discrimination

Iowa State University strives to maintain our campus as a place of work and study for faculty, staff, and students that is free of all forms of prohibited discrimination and harassment based upon race, ethnicity, sex (including sexual assault), pregnancy, color, religion, national origin, physical or mental disability, age, marital status, sexual orientation, gender identity, genetic information, or status as a U.S. veteran. Any student who has concerns about such behavior should contact his/her instructor, Student Assistance at 515-294-1020 or email `dso-sas@iastate.edu`, or the Office of Equal Opportunity and Compliance at 515-294-7612.

Religious Accommodation

If an academic or work requirement conflicts with your religious practices and/or observances, you may request reasonable accommodations. Your request must be in writing, and your instructor or supervisor will review the request. You or your instructor may also seek assistance from the Dean of Students Office or the Office of Equal Opportunity and Compliance.