
Illinois State University

ITK 179, Fall 2006

Introduction to Data Structures

Stevenson 139A, MW 12:35 ~ 1:50 PM

Instructor: Chung-Chih Li, Ph.D.

<http://www.itk.ilstu.edu/faculty/chungli>
cli2@ilstu.edu

Office: Old Union 105, Tel:(309) 438-7952

Office Hours: MTWT 2:30 ~ 3:30 PM or by appointment

Classroom and meeting time: (Attendance will be taken impulsively)

Stevenson 139A, MW 12:35 ~ 1:50 PM

WebPage of the course: <http://www.itk.ilstu.edu/faculty/chungli/ITK179>

Students should check the Webpage of the class from time to time. From there, you may find important information about assignments, data, due dates, sample programs, or announcements.

Note: *An announcement made in the class will be considered as an official one, since I may not be able to update every announcement.*

Prerequisites:

ITK 168 with a C or better grade.

This is a continuous course of ITK 168. We assume that students already have basic skill in developing easy JAVA programs including using some IDE, primitive data types, control structures, and basic I/O of the language.

Note: *Students will not receive credit if they have taken ITK 169 or ITK 275.*

Textbooks:

Objects, Abstraction, Data Structures and Design Using JAVA, Version 5.0, by Elliot B. Koffman & Paul A.T. Wolfgang, Wiley 2005.

Reference:

Java, Software Solution – foundations of program design, 4th edition, by John Lewis and William Loftus, Addison Wesley, 2004

Course Description and Purposes:

In this course, we will further strengthen your programming skill and learn more problem-solving techniques through studying some advanced features and concepts of the languages. We will emphasize on both traditional data structures and the relatively new programming paradigm – object-oriented design.

The data structures we are going to learn include *linked list, stacks, queues, trees, and binary search trees*. We will organize our data into those structures and see how those data structures improve our programs' performance. We will also learn how to measure the computational complexity of a program or a particular problem. Also, we will learn a powerful programming technique – *recursion* – and understand it as a natural approach to solving complicated problems.

We assume you already have some basic concepts of Java programming from ITK 168. Nevertheless, at the beginning of the semester, we will briefly review the basic yet necessary features

of Java and explore you to more Java IDE including Eclipse and NetBeans. We will scratch the surface of UNIX and run Java programs from the *command line* mode as the default programming environment throughout the semester. Issuing commands under *command line* to operate an OS is primitive but yet the most direct way to use the OS, since working under the command line mode can minimize the task run behind the scenes. Thus, we encourage you to develop and test your Java programs under the UNIX command line mode and be familiar with the basic UNIX's commands for your feature study. However, you can use any IDE to develop your programming assignment.

Objectives of the Course: After this course, students should:

1. Understand and be able to implement the basic data structures: *linked list, stacks, queues, trees, and binary search trees.*
2. Understand recursion and be able to solve some typical problems recursively.
3. Be able to design, implement, test, and debug moderately complex Java programs.
4. Be familiar with the paradigm of Object Oriented Programming: *Classes, objects, methods, class inheritance and hierarchies, and polymorphism.*
5. Be ready to study more advanced algorithm analysis.
6. Be comfortable to explore to any new IDE in the feature.
7. Be comfortable to explore to Unix in the feature.

Examinations: (500 points) Three midterms and one final exam; 100 points for each midterm and 200 points for the final.

- Unless announced otherwise, all tests are accumulative, closed book, and indispensable. No makeup test will be given unless a documented absence is authorized by the university.
- Every student is allowed to bring a *self-prepared hand-writing* crib sheet to the test. You can **write** down anything on both sides of **one** letter-sized paper. No circulation during the test.

4 th week's	Midterm I	100 points	Sep. 11, Monday
8 th week's	Midterm II	100 points	Oct. 9, Monday
12 th week's	Midterm III	100 points	Nov. 6, Monday
17 th week's	Final Exam	200 points	Dec. 12, Tuesday, 1:00 PM

Assignments: (400 points) About 7 or 8 programming assignments will be given. The weight of each assignment depends on the difficulty of the assignment; usually between 30 and 60 points.

Programming assignments should be prepared according to the following requirements for submission. **Read them carefully; they are applied to every programming assignment!**

1. Put a few comment lines at the beginning of your program file, in which you should clearly indicate your name and ID and claim your copyright. **Student who fails to do so will receive 0 point on the assignment.**
2. Submit your assignment with items in the following order:
 - (a) A cover page with your name and student ID on it.

- (b) A brief summary about the assignment and your approach to the problem. You may include the difficulties you had faced, if any, or why you think your program doesn't work. It is very common and not a shame to admit that your program doesn't work under time constraints, but a reasonable self-diagnosis deserves reasonable partial credit.
- (c) A hard-copy of the source codes.
- (d) A hard-copy of the directly output of your program, if any.
- (e) A diskette, R/W CD, or USB memory stick containing all source and byte codes *under the root directory*. Don't use DVD.
- (f) All item above should be put in a letter-sized Manila folder with your name on it.

All materials will be returned after graded, and you are encouraged to reuse them, if appropriate. However, you should prepare at least two sets, because I will not be able return your work before your next assignment begins.

Do backup your works as often as possible. Remember: bad things do happen, and "my dog ate my works" is not a good excuse.

Students are encouraged to discuss assignments and help each other. However, this does not mean that you can either entirely or partially copy or modify other's works.

Try very hard to avoid the following troubles:

1. Any form and any degree of plagiarism will receive 0 point.
2. If your program contains syntax error, i.e., I can't compile the program, you will receive 0 point. Note well that the difference between "my program can't be compiled" and "my program doesn't work" is huge. The former one deserves nothing.
3. If the hard-copy of the direct output of your program is inconsistent to your program's design, you will receive 0 point. This is a kind of cheating.
4. Late works will be graded with penalty: -10 points per day after the due date. A weekend is counted as 2 days.

Attendance: Attendances will be taken impulsively. Each unauthorized absence will cost you 50 points from your score tally.

Pop quizzes: (100+ points)

Some pop quizzes will be given without notice in advance. Each quiz carries 20 points towards students' final scores. The coverage of every quiz is also accumulative, including the materials that are three-month-old and those covered in the class right before the quiz. A typical quiz takes about 10 minutes. No makeup quiz will be given if missed. If you miss a quiz due to a university authorized absence, we will use the average of your rest quizzes as the score; otherwise, you get a 0 for the missing quiz.

Academic Honesty:

Cheating, plagiarism, collusion, abuse of resource materials, and their consequences are defined and described in ISU 2006-2007 Undergraduate Catalog, Section: Academic Policies and Practices, Article: Academic Integrity (Page 63) and Code of Student Conducts under X.C. Disciplinary Bodies And Procedures – Academic Honesty Cases .

Students giving away academic works for assignment offered for credit to other students working on the same assignment will be considered as guilty as academic dishonesty, and will receive the same penalty.

More information can be found at:

http://www.deanofstudents.ilstu.edu/crr/downloads/Code_of_Student_Conduct.pdf

Grading Policy:

If there are more than five quizzes given throughout the semester or if we think some assignments are difficult enough to deserve more points, you may have more than 1000 points to gain. But we always consider 1000 points as the perfect score. your grade is based on the scheme shown in the following table.

Points	Grade	
850 ~ up	A	Excellent
750 ~ 849	B	Good
600 ~ 749	C	Satisfactory
500 ~ 599	D	Passing
0 ~ 499	F	Failure

I do not curve!!

I'm not afraid to give all A's, neither am I to give all F's. In other words, you don't have to knock down your friends to get a good grade, so do help them if they need you, and you can't hide behind someone else, because you two could be both shot down.

Tentative Topics and Schedule:

Keep the table of tentative topics and schedule in the following page handy, and try to keep up with the schedule. Read the assigned materials before the class.

Tentative Topics and Schedule for ITK 179, Fall 2006

Week	Topics	Reading
0: –	Kicking off	Syllabus
1: Aug. 21	Java preliminaries, JDK, IDE, Eclipse, NetBeans, Unix	Handouts, A.1~A.3
2: Aug. 28	Software Developing, Testing, Debugging, and Efficiency Analysis	Chapters 1, 2
3: Sep. 4	OOP (Object-Oriented Programming) and OOD (Object-Oriented Design), ArrayList Classes	Chapter 3, 4.1~4.3
4: Sep. 11	(Midterm 1) Single-Linked Lists and Double-Linked Lists	4.4, 4.5
5: Sep. 18	Double-linked lists and their implementations and applications	4.6~4.8
6: Sep. 25	Stacks and their implementations and applications	5.1~5.3
7: Oct. 2	More on Stacks, Queues	5.4, 6.1, 6.2
8: Oct. 9	(Midterm 2) Queue implementations	6.3, 6.4
9: Oct. 16	Recursion in mathematics and programming languages	7.4~7.6
10: Oct. 23	Recursive Data Structures, Backtracking	7.4~7.6
11: Oct. 30	Trees and their traversals, Binary Trees	8.1~8.3
12: Nov. 6	(Midterm 3) Binary Search Trees, Priority Queue	8.4, 8.5
13: Nov. 13	Basic quadratic sorting algorithms and their efficiency	10.1~10.6
14: Nov. 20	Thanksgiving break	review/catch up
15: Nov. 27	Merge Sort, Heap Sort, Quick Sort	10.7~10.9
16: Dec. 4	Tree Balance and Rotation, AVL Trees	11.1, 11.2
17: Dec. 11	Final Examination, Dec. 12, Tuesday, 1:00 PM	all

Play Ball!!