Comp 270

Data Structures

Spring, 2018

Day/Time:	Tuesday, 5:30 - 8:30
Prerequisites:	COMP 170
Instructor:	Dr. Channah Naiman
email:	naiman@htc.edu

Syllabus Index	HTC Policies

Syllabus Index

Objectives	Exams
Texts and Software	Grades
Class Format, Attendance	Programming Environment
Homework/Assignments	Course Schedule
Project	

HTC Policies

Accommodations	Attendance
Academic Integrity	Incomplete Policy
Technology Policy	HTC Resources
Conceptual Framework	Mid-semester Progress Reports
Course Evaluations	

Conceptual Framework

ללמד וללמד, לשמר ולעשות

"To Learn, To Teach, To Observe, and To Do." The student should be a *Critical Thinker, Effective Communicator, Proactive Educator, and Moral Practitioner.*

Course Description:

This continuation of COMP 170: Introduction to Object-Oriented Programming introduces key data structures such as lists, sets, maps, queues and stacks, as well as their implementation. Performance and analysis of algorithms are covered, along with applications in sorting and searching.

Outcome:

Students will learn to design new data structures as well as learn to

use existing data structures in applications.

This course is programming-intensive. It includes several major

programming projects, as well as a weekly lab component.

Additional time working on reading, preparing, and programming will be

required outside of class time.

Prerequisites:

Prerequisite: COMP 170
Corequisite: COMP 163
Stuents are exptect to be able to read, write and ebug basic computer programs using standard tools incluidng compilers and editors.
Students are expected to be able to read,write and debug basic computer programs using standard tools including compilers and editors
Students are expected to know the basics of Object-Oriented Proramming and be able to use classes and methods they or others have written.
The logical mindset of mathematics in generally helpful in learning programming. See me if you have any questions or concenrs about your preparation.

Course Objectives:

Upon successful completion of the course, the student will be able to:

1. Discuss the appropriate use of built-in data structures. Describe common applicaitons and construct and debug programs that use each of the following data structures: lists, stacks, queues, trees, sets, maps and graphs.
2. Describe the concept of abstract data types (ADT) in terms of interface (contrac), client (user) and implementation (provider). Explain how to implement abstract data types in an object-oriented language using classes, arrays and linked structures.
3. Describe the concept of recursion and give examples of its use. Implement, test and ebug simple recursive functions and procedures. Determine whether a recursive or iterative solution is most appropriate for a problem.
4. Explain why the creation of correct program components is important in the production of high-quality software. Describe how a contract can be used to specify the behavior of a program component. Apply a variety of strategies to testing and debugging simple programs.
5. Analyze the extent to which another programmer's code meets documentation and programming style standards.

NOTE: There are additional skills that will be covered that are not related to Data Structures, but rather are useful for all professional software developers. These include the use of an IDE (IntelliJ in our case), and Version Control Software (Git, and hosting on Github).

Course Materials:

Data Structures: Abstraction and Design Using Java.

E.B.Koffman and A.T.Wolfgang. Wiley 2015 3rd

edition. ISBN 978-1-119-18652-6. Online companion site (freely accessible)

The online version is recommended. You can try the online text free for 14 days at www.WileyStudentChoice.com

Additional course materials will be online, on Blackboard, or provided in class.

The official documentation for Java version used in class is from Oracle

https://docs.oracle.com/javase/9/docs/api/

It is useful to look up specific terms, concepts, and other things you want to see.

Oracle has a much material to help Java programmers: start from this link:
http://www.oracle.com/technetwork/topics/newtojava/overview/index.html

Course Format:

The course is structured around 10 labs. The labs include the text material, as well as other material covered in class or included as links. In general the labs include a starter template with comments inserted instructing you what you need to do to complete the code. And in general, the labs are written in a more sophisticated level of code than you would normally be expected to write at this level, incorporating concepts and techniques that are not, strictly speaking, part of a data structure assignment. So in this way, you get the exposure to a better way to code without having to be fully responsible for writing this code from scratch. The first lab is a very simple Java program (something you should find very easy if you've taken COMP 170). However, you will be required to complete the Orientation activities, including installing software, configuring git, using IntelliJ, importing a gradle-wrapped Java project from GitHub to IntelliJ, and other "housekeeping" type of activities. Often, the labs will require that you reference a structure that already exists in Java, but you are required to write your own version of the structure. In this way, you will become very familiar with thecapabilities and limitations of the structures.

There is recent evidence that with all the online resources available, a hybrid approach is useful for on-campus courses. We will occasionally have a "fippled" class. This means that you get most of your presentations and do much of your homework, so the text/videos are mostly at your convenience, at your speed, and then in class discuss questions you had on the presentations and do much of the harder creative work of synthesizing and using this information, when you have the most direct support from me. I will probably start with more lecture, and as you get more comfortable with the ebook and videos, flip the class more as we progress through the course. This is especially the case for the project, where you will want more individual help from me. Videos will be posted as they become available.

Homeworks/Assignments:

The labs are also your homeworks. While you may often begin them during class time, you will almost certainly need significant additional time outside of class to complete them.

Pair Programming: It has been demonstrated recently that Pair Programming, two people collaborating on one problem with one person coding while the other looks on, whether beginner students or seasoned professionals, allows projects been done better and faster with more confidence, and also that students learn at least as well and have more enjoyment in the process. We will have the option to do pair programming in this course for in-class work and programing assignments. (Your exams will NOT be in pairs however!) Read the page on how to make pair programming work and also the page of administrative guidelines for pair programming (mostly for when it does not work out as planned!).

Project: You will form teams of two or three, to complete a project, which is a major programming assignment in JAVA. Some suggestions for the project can be found here. This will be discussed further in class.

Programming Environment:

We will be using an infrastructure including Java 9, IntelliJ, gradle, git/GitHub.   See the installation and configuration instructions in the Orientation module on Blackboard. We will also be using JUnit Testing, which is described in Module 1 on Blackboard.

Project:

There is a final project that requires students to work in teams to create an original implementation of an application that incorporates the use of data structures.

Exams: There are two exams scheduled, one on 3/20, and one on 6/05.   I reserve the right to cancel the final exam, if, in my judgment, your Final Projects demonstrate sufficient mastery of the material. Don't count on it--the projects have to be truly excellent for me to do that, but it has happened on rare occasion.

Exams will any cover material discussed in class, and any programming techniques covered in the labs, as well as the textbook reading assignments.   You are allowed two 8.5 x 11 inch sides of notes for exams, but no computer or calculator. I emphasize having you process and use information, not regurgitate facts -- put the facts you most forget and still need in your notes. This is very different than many of the requirements some students had in high school, where fact recall may have been key. What you want most to remember is general patterns about the process of breaking problems into pieces and identifying the right process for each piece. The pieces may come in all sorts of combinations, so remembering whole rote sequences is not likely to be helpful. Do not depend on it.

There will be a review for the exams posted on Blackboard.

Grading:
Your grade will consist of several components with relative weights as follows. (I reserver the right to adjust the percentages in your facor if circumstances warrant.

Orientation and Installation	50
Lab/Homeworks All labs are worth 30 points.	300
Exam (Midterm Exam: 200 points) (Final Exam: 200 points) If there is no Final Exam, the midterm will be worth 300 points.	400
Project If there is no Final Exam, the project will be worth 300 points.	200
Participation/Attendance	50
Total	**1000**

- There is no extra credit. It is not practical, nor is it fair to other students in the class. At my discretion, I may assign an extra credit assignment to the entire class.
- Late assignments are worth only half credit. This is true even if you have a valid reason for submitting the homework late.   Late assginments must be submitted within one week of the due date for half credit. After one week, you will receive zero points for any unsubmitted assignments.
- You may only have two late submissions. After you use up your two late submissions, anything that you submit after the due date is worth a zero (not half credit).

Course grades are assigned as follows:

A+	97

A	93
A-	90
B+	87
B	83
B-	80
C+	77
C	73
C-	70
D+	67
D	63

Course Schedule:

The dates below give the sequence and a general idea of the time spent, though we may get ahead or behind this time schedule at different points, depending on the needs of the class. Every attempt will be made to meet this timeline, and changes will be posted on Sakai. Links for supporting docuements and files, and submission links for assignments are on Blackboard.

Week	Date	Topic	Readings/Materials	Assignments
0		Orientation (before class begins)	• Installtion/configuration instructions on Blackboard • Videos to install, to create GitHub account, to clone using git • Videos to import into IntelliJ	Installation, Configuration and import assignments Details on Blackboard
1	1/30	• Orientation if not complete • Java Review • Infrastructure Review	• Appendices A.1 - 3; A.6 - 8 • Chapter 1.1 - 1.6 (PPTs posted on BB) • Java Review [slides 1-46] • Java programming “cheat sheet” [link] • Example: hello-java	Lab 1: Infrastructure and FizzBuzz
2	2/06	• More Java Review • Command-line args • Basic I/O • ArrayLists (in passing) • JUnit testing	• Chapter 2.2-2.4; 3.1 - 3.7 (PPTs on BB) • Videos on JUnit Testing	Lab 1, continued
3	2/13	• Quantifying Algorithms (Big-O) • Searching: linear, binary	• Chapter 2.1 • Big - Slides [slides] Searching: linear, binary	Lab 2
4	2/20	• ArrayList Implementation • LInked List	• Chapter 2.2 - 2.5	Lab 2, continued
5	2/27	• Linked lists in more detail	• Chapter 2.5 - 2.6	Lab 3
6	3/06	• Iterators	• Chapter 2.7 - 2.10	Lab 4
7	3/13	• Stacks	• Chapter 4.1 - 4.4	Lab 5
8	3/20	• Queues • Midterm Exam	• Chapter 4.5 - all	Lab 6
9	4/10	• Recursion	• Chapter 5.1 - 5.3	Lab 7

10	4/17	• Recursion, Cont'd • Sets and Maps (begin)	• Chapter 5.4 - 5.5 • Chapter 7.1 - 7.3	Lab 7
11	4/24	• Sets and Maps, cont'd • Implementation using hash tables	• Chapter 7.4 - 7.6; Chapter 8.1	Lab 9 CUE: Project Proposal
12	5/01	• Sorting	• Chapter 8	Lab 9, cont'd
13	5/08	• catch up		
14	5/15	• Graphs	• Chapter 10	Lab 10
15	5/29	• Graphs, cont'd		
16	6/05	Final Exam and Project Presentations		

HTC Policies

Attendance and Participation

1. Students are required to attend, be prepared for, and actively participate in all classes.
2. Students are required to attend all classes. In the case of absences, a student must communicate with instructor immediately via email.
3. Students absent for over 25% of scheduled classes will have their final grade lowered by one letter grade.
4. Students absent for over 50% of scheduled classes will receive an F (failure) in the course.

Please note that without extenuating circumstances, this policy may not be changed by the instructor. If you have any problems or concerns, please see Dr. Tessler.

Academic Integrity

Hebrew Theological College is committed to providing an academic community and learning environment based on honest inquiry and pursuit of knowledge that fosters commitment and adherence to Judaic tenets. The faculty and administration of Hebrew Theological College have specified the following acts as serious violations of personal honesty and academic ideals that jeopardize the quality of education within a Torah environment:

- Submitting as one's own, material copied from a published source.
- Submitting as one's own, another person's unpublished work or examination material.
- Submitting as one's own, a rewritten or paraphrased version of another person's work.
- Purchasing, acquiring, and using for course credit a pre-written paper.
- Allowing another to write or research a paper for one's own benefit.
- Copying electronic or printed media for one's own use without permission or licensing from appropriate publishers.
- Submitting the same paper for more than one course without explicit permission from the instructor(s).

More information about HTC’s Academic Integrity policy can be found on page 15 of the Student Handbook.

Accommodations

Any student, who, because of a disability, may require some special arrangements in order to meet course requirements should contact the instructor as soon as possible to make necessary accommodations and share appropriate documentation from the Office of Special Services, provided by HTC’s Disabilities Officer, Dr. Richard Aronoff.

Accommodations will be made, but instructors must be aware of your needs in order to make proper accommodations. It is the responsibility of the student to make these needs known in a timely fashion and to provide documentation prior to the beginning of any semester in which accommodations are desired.

Incomplete Policy-- Crisis Management

This is available ONLY to students with extreme and/or extenuating circumstances who

1. have completed 50% or more of the required course work.
2. have a grade of "C" or better on completed work.
3. request the "Incomplete" prior to the week of final examinations of the semester.
4. complete and submit an "Incomplete Contract" prior to final examinations.
5. The "Incomplete Contract" must be signed by the course instructor and the Dean. Non-compliance by agreed date will result in a permanent grade of FI (Failure/Incomplete).

Students who have not satisfactorily completed 50% of class assignments are not allowed to receive an incomplete grade. Students who do not have a grade of “C” or above average are not allowed an incomplete. Students must request an incomplete prior to the week of finals. The instructor may refuse an incomplete request. It is the student's responsibility to request an Incomplete Form from the school office, have the form signed by the instructor, Dean, themselves, and return it to the instructor prior to the final. If the student fails to return the completed form to you when grades are due, the student is to receive the grade they would have received if they had not requested an incomplete. Instructors will define the amount of time allowed to complete the missing work, however, the time allowed cannot exceed 12 weeks after the end of the semester in which the incomplete was requested. If the student fails to remove the incomplete within 12 weeks, the grade will be recorded as an "IF," meaning failure due to an incomplete. An "IF" will be counted as an "F" in the student's GPA and will be part of the student's permanent record.

Technology Policy

HTC bans the use of cell phones, computers and other devices for texting,

web-browsing or other non-class related activities during class. This behavior may result in expulsion from the course after a single warning. Electronic devices may not be used during exams, and their use in class is subject to faculty discretion and permission. Only students with documented disabilities who must use such devices may request exemptions as documented.

HTC Academic Resources

Hebrew Theological College is committed to providing all of our students with various resources and support for academic success. Tutorial services through the Writing Clinic, Math Center, and Hebrew Tutoring Center provide assistance in a variety of disciplines. Students should make arrangements to avail themselves of these services. Librarians at the Saul Silber Memorial Library are available to assist students with all their research needs. Students can find information about the library services and resources at http://htclibrary.weebly.com.

Mid-Semester Progress Reports

Mid-way through the semester, your instructor will evaluate your progress in this course and share this feedback with the deans. This information will be used for advising purposes. You can view the form at http://tinyurl.com/HTCMSPR.

HTC Course Evaluations

In order to help make the course evaluation process more convenient and ensure student privacy, you can now submit secure, anonymous course evaluations online via LiveText! Your instructors will provide time in class to complete the form, but you can also fill it out on your own at any time. Your feedback will be shared with your instructors after grades have been turned in. Instructors are eager to know how they can improve and rely on your feedback as a central part of their professional development. This is your opportunity to express your thoughts about your experiences at Hebrew Theological College. Your voice matters.