JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Syllabus
Computer Science 600.226
Data Structures
Fall 2018 (4 credits, EQ)

Instructors

Associate Professor Michael C. Schatz, mschatz@cs.jhu.edu,
http://www.cs.jhu.edu/~mschatz/
Office: Malone 323
Office hours: Wednesday, 3:00-4:00 pm; and by appointment

Teaching Assistant

To be announced on course website

Meetings

- Monday, Wednesday, Friday, 1:30–2:45 pm in Mudd 26.

Textbook

No textbooks are required but one is recommended:

- Clifford A. Shaffer, *Data Structures and Algorithm Analysis (Java Version)*: on-line interactive JHU version (OpenDSA) available at http://algoviz.org/OpenDSA/Books/CS226JHUS16; (older) print edition 3.2 available at http://people.cs.vt.edu/shaffer/Book/JAVA3elatest.pdf and through Dover Publications.

The three titles below are recommended as additional reference options.

- Mark Allen Weiss, *Data Structures and Algorithm Analysis in Java (3rd edition)*, Pearson Education (2012). ISBN: 9780132576277.
- Robert Sedgewick and Kevin Wayne, *Algorithms (4th edition)*, Addison-Wesley (2011). ISBN: 9780321573513. The JHU Library has an online version of this text that each of you should be able to access for free.
- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to Algorithms (3rd edition)*, Massachusetts Institute of Technology (2009). ISBN: 9780262033848.This text is often required for EN.600.363 Introduction to Algorithms.

For students who feel they need resources about the Java programming language beyond what is available on the Internet, the following books are suggested.

- John Dean and Raymond Dean, *Introduction to Programming with Java: A Problem Solving Approach (2nd edition)*, McGraw-Hill (2014). ISBN-13: 978-0073376066. This text has recently served as the required text for EN.600.107 Introductory Programming in Java.
- Evans and Flanagan, *Java in a Nutshell (6th edition)*, O'Reilly (2014). ISBN-13: 978-1449370824.
- Deitel and Deitel, *Java How to Program (10th edition),* Prentice-Hall (2014). ISBN-13: 978-0133807806
- Arnold, Gosling, and Holmes, *The Java Programming Language (5th edition)*, Addison-Wesley Professional (2013). ISBN-13: 978-0132761680

Online Resources
The following online resources are essential:

- The course web site location at https://github.com/schatzlab/datastructures2018. You will find a schedule of topics, class notes, and assignment details there.
- The course Piazza site at https://piazza.com/jhu/fall2018/600226/home. This site will serve as our discussion site for the course. Please use Piazza to ask questions of the course staff and fellow students.

Course Information

- This course covers the design and implementation of data structures including arrays, stacks, queues, linked lists, binary trees, heaps, balanced trees (e.g. 2-3 trees, AVL-trees) and graphs. Other topics include sorting, hashing, memory allocation, and garbage collection. Course work involves both written homework and Java programming assignments.

Course Goals

Upon successful completion of this course, you should be able to:

1. Evaluate and compare the time complexity of functions using mathematical techniques.
2. Design an algorithm that produces the correct results according to specified inputs and time or space complexity constraints.
3. Understand the operation of common data structures and algorithms.
4. Use analysis techniques to choose the data structure/implementation appropriate for a given problem.
5. Write advanced object-oriented solutions in Java to significant problems, by implementing appropriate data structures and algorithms.

This course will address the following Criterion 3 Student Outcomes

- An ability to apply knowledge of computing and mathematics appropriate to the discipline (a)
- An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution (b)

- An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs (c)
- An ability to function effectively on teams to accomplish a common goal (d)
- An ability to use current techniques, skills, and tools necessary for computing practice (i)
- An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices (j)
- An ability to apply design and development principles in the construction of software systems of varying complexity (k)

Course Topics

The goal of the course is to teach fundamental data structures, which allow one to store collections of data with fast updates and queries. Key topics will definitely include: Java refresher and generics, analysis tools, sorting, linked lists and iterators, stacks and queues, search trees, maps, hashing, priority queues, and graphs. Please see the main course website for a more detailed schedule, which will be updated as the semester progresses

Course Expectations & Grading

Course grades will be based on assignments (typically Java implementations), a midterm, and a final, according to the proportions below. Each homework assignment will be assigned a point value; the overall homework assignment grade will be computed as your total points earned divided by the total achieved in the class.

- 50% - Assignments (some individual, some with others)
- 20% - Midterm (Oct 12, 2016 within class)
- 30% - Final Exam (TBD)

All grades will be distributed via email. Please keep your own record of your grades so that you will know your standing in the course. Letter grades for the course will be assigned on a standard scale, subject to the instructor's evaluation of your overall class performance. Do not expect a curve in this course.

Assignment Logistics. The implementation projects in this course will require you to design and write Java programs that compile with the standard Java 8 compiler and run on the virtual machine that we distribute. You will receive no credit for programs that do not compile on this virtual machines. In addition, all students will be provided a Linux account from the Computer Science Department, and will be given access to the CS Undergraduate Lab in Malone Hall 122 to work on assignments and meet with course assistants.

You are also free to download a Java compiler and do your work on your own computer, though it is recommended that you use either a Unix/Linux environment or an integrated development environment such as Eclipse for your work in this course. If working on

your own machine, you should always check that your code compiles and runs correctly on the virtual machine since assignments will be graded there.

Attendance. All students are generally expected to attend all meetings of this course, and actively participate in all course meetings. If you miss a class meeting for any reason, you are responsible for material presented, and it is your responsibility to obtain any missed handouts or other materials.

Grace Period. For grade items that come with a specific deadline (most commonly projects and assignments) there is a grace period of one hour: If you submit within one hour after the posted deadline, we will still accept your submission but your total score for the grade item in question will be reduced by 10%.

Grace periods interact with late days in ways that may not be completely obvious at first, so here's an example. Say an assignment is due at 10:00 pm, and let's say that your submission would have netted you 90 points. There are the following possibilities:

- If you submit before 10:00 pm, up to and including 10:00 pm itself, you would receive 90 points.
- If you submit after 10:00 pm but before the grace period is over, so anytime between 10:01 pm and 11:00 pm, you would receive 81 points instead (a penalty of 9 points).
- If you submit after 11:00 pm and you still have a late day left, then you use up one late day and your new deadline is 24 hours after the original deadline; so in this example the new deadline is 10:00 pm the next day; that new deadline comes with a new grace period as before.
- If you submit after 11:00 pm and you don't have a late day left, then you will receive 0 points (just as if you had never submitted the assignment).

Note that each 24 hour period that you are late with your submission will eat up a late day, regardless of whether or not you have enough total late days. So if you have 2 late days left, but you submit at a time when you would have needed 3 late days, both your 2 late days are gone and your submission receives 0 points. Be very careful about when exactly you submit something!

Late Days. All students start the semester with a budget of five late days. A late day can be used to extend the deadline of a grade item by exactly 24 hours. Late days cannot be used to delay an exam or a presentation. Late days cannot be used to make up attendance. Late days have to be used in their entirety. You can only use a late day if you have one left in your budget; a team can only use a late day if all team members still have a late day left.

All you need to do in order to use a late day is submit a grade item after both the deadline and the grace period are expired. There is no need to "request" a late day in advance, it's solely based on when you submit. Of course you must still have a late day available, otherwise the late submission will result in a score of 0. Note that since late days are used up automatically based on your submission, it's very important that you weigh your

options carefully. You cannot "get back" a late day after the fact because you (or one of your team mates) made a mistake.

Late days are a valuable commodity and you should use them sparingly. They are primarily intended to help you deal with unexpected circumstances, so you should not make them part of your normal planning process. (Also see Illness and Religious Holidays.)

Note: You cannot use late days to delay an assignment until after the semester is over (meaning after the final exam for your course has been given). You can, however, delay an assignment that's due on the last day of classes into reading period; you do this at your own risk, we are not responsible when you do worse on other exams as a result of spending your reading period working on an overdue assignment.

Key Dates
Assignment and exam schedule will be distributed on the course webpage.

Assignments & Readings
Assignment and exam schedule will be distributed on the course webpage.

Ethics
The strength of the university depends on academic and personal integrity. In this course, you must be honest and truthful, abiding by the *Computer Science Academic Integrity Policy*:

> Cheating is wrong. Cheating hurts our community by undermining academic integrity, creating mistrust, and fostering unfair competition. The university will punish cheaters with failure on an assignment, failure in a course, permanent transcript notation, suspension, and/or expulsion. Offenses may be reported to medical, law or other professional or graduate schools when a cheater applies.
>
> Violations can include cheating on exams, plagiarism, reuse of assignments without permission, improper use of the Internet and electronic devices, unauthorized collaboration, alteration of graded assignments, forgery and falsification, lying, facilitating academic dishonesty, and unfair competition. Ignorance of these rules is not an excuse.
>
> Academic honesty is required in all work you submit to be graded. Except where the instructor specifies group work, you must solve all homework and programming assignments without the help of others. For example, you must not look at anyone else's solutions (including program code) to your homework problems. However, you may discuss assignment specifications (not solutions) with others to be sure you understand what is required by the assignment.
>
> *If* your instructor permits using fragments of source code from outside sources, such as your textbook or on-line resources, you must properly cite the source. Not citing it constitutes plagiarism. Similarly, your group projects must list everyone who participated.

Falsifying program output or results is prohibited.

Your instructor is free to override parts of this policy for particular assignments. To protect yourself: (1) Ask the instructor if you are not sure what is permissible. (2) Seek help from the instructor, TA or CAs, as you are always encouraged to do, rather than from other students. (3) Cite any questionable sources of help you may have received.

On every exam, you will sign the following pledge: "I agree to complete this exam without unauthorized assistance from any person, materials or device. [Signed and dated]". Your course instructors will let you know where to find copies of old exams, if they are available.

In addition, the specific ethics guidelines for this course are:

1. In the completion of individual homework assignments, you may not discuss your approach with or show specifics of your code to others. This includes fellow students, former students, friends, etc. You are permitted to request assistance from course staff (instructors, TAs and CAs) only.
2. You are permitted and expected to reuse and adapt code from lectures and the assigned text (Shaffer - either version) in completing your projects. However, all original sources must be cited in comments within your code.
3. In using Piazza to ask questions about homework assignments, you should post privately to Instructors any questions that involve code or that would give away your approach to solving the assignment. Otherwise, you are encouraged to ask general, abstract questions, and post them publicly, so other students may benefit from from the discussion.

Report any violations you witness to the instructor.

You can find more information about university misconduct policies on the web at these sites:

- For undergraduates: http://e-catalog.jhu.edu/undergrad-students/student-life-policies/
- For graduate students: http://e-catalog.jhu.edu/grad-students/graduate-specific-policies/

Students with Disabilities
Any student with a disability who may need accommodations in this class must obtain an accommodation letter from Student Disability Services, 385 Garland, (410) 516-4720, studentdisabilityservices@jhu.edu .