JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Syllabus
Computer Science 601.226
Data Structures
Summary 2019 (4 credits, EQ)

Instructors

Gabe Kaptchuk, gkaptchuk@jhu.edu
https://cs.jhu.edu/~gkaptch1
Office hours: Malone 307, hours to be posted on Piazza

Joanne Selinski, joanne@cs.jhu.edu
https://cs.jhu.edu/~joanne
Office hours: Malone 225, hours to be posted on Piazza

Teaching Assistants

Erin Chen, echen41@jhu.edu
Office Hours: Malone 122, hours to be posted on Piazza

Meetings

Monday, Tuesday, Thursday, Friday 1-4:15 pm in Hackerman B17

Textbooks

Required (free):

- Clifford A. Shaffer, *Data Structures and Algorithm Analysis (Java Version)*: on-line interactive text available at https://opendsa.cs.vt.edu/ODSA/Books/Everything/html
  - We will not use all of this material, but specific readings will be assigned each week.
  - (older) print edition 3.2 available at http://people.cs.vt.edu/shaffer/Book/JAVA3elatest.pdf

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.601.463 Intro 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*, McGraw-Hill
- Evans and Flanagan, *Java in a Nutshell*, O'Reilly
- Deitel and Deitel, *Java How to Program,* Prentice-Hall
- Arnold, Gosling, and Holmes, *The Java Programming Language*, Addison-Wesley Professional

Online Resources

The following online resources are essential:

- The course Piazza site at https://piazza.com/jhu/summer2019/601226/home. This site will serve as our discussion site for the course, as well as the location for all documents (general resources, assignments, lecture notes, etc.). Please use Piazza to ask questions of the course staff and fellow students.
- Gradescope will be used for all homework submission, grading and feedback. (You'll receive an invitation.)

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 Program Student Outcomes

- Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions. [SO1]
- Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline. [SO2]
- Apply computer science theory and software development fundamentals to produce computing-based solutions. [SO6]

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, Junit testing, analysis tools, sorting, linked lists and iterators, stacks and queues, search trees, maps, hashing, priority queues, and graphs. Please see the course website for a more detailed schedule, which may be updated as the semester progresses.

Course Approach

This course will be taught with an *active learning approach*, meaning that class sessions will be a mixture of lecture content and student activities intended to help you understand and digest the material. Students are expected to look at assigned reading materials before class each week in order to make the most of class time. Staff will be on-hand to assist with activities. We will not be tracking participation for grading purposes. However, you are strongly recommended to take full advantage of this approach to learning.

Course Expectations & Grading

Course grades will be based on assignments (typically Java implementations), two midterms, 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 achievable points.

- 0% - Class participation (expected, but not tracked)
- 50% - Assignments
- 15% - Midterm 1 (Friday, 7/12)
- 15% - Midterm 2 (Monday, 7/22)
- 20% - Final Exam (Friday, August 2 – last day of class)

All grades will be distributed via Gradescope. Letter grades for the course will be subject to the instructor's evaluation of your overall class performance, generally based on this standard scale: >97 A+, 93-97 A, 90-93 A-, 87-89 B+, 83-86 B, 80-82 B-, 77-79 C+, 73-76 C, 70-72 C-, 67-69 D+, 60-66 D, <60 F. 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 Gradescope. You will receive no credit for programs that do not compile. You are expected to download the Java compiler and do your work on your own computer. It is recommended that you use either a Unix/Linux environment or an integrated development environment for your work in this course, and detailed information will be provided in our resources section on Piazza. By request, students may be provided a Linux account from the Computer Science Department, and given access to the CS Undergraduate Lab in Malone Hall 122 to work on assignments and meet with course assistants. See the instructor for a signature on the account request form if you don't already have this access.

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. There is no grace period for assignment deadlines, only late days.

Late Days. All students start the semester with a budget of five late days; however you can only use at most 1 late day for any particular deadline/assignment. Each late day extends the deadline of a grade item by exactly 24 hours. Late days cannot be used to delay an exam or a presentation. 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 the deadline has expired through the regular submission. 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 teammates) 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.

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 show specifics of your code to others or copy code from other sources or students. This includes fellow students, former students, friends, etc. You are encouraged to request assistance from course staff (instructors, TAs and CAs).
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 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: https://studentaffairs.jhu.edu/policies-guidelines/undergrad-ethics/
- For graduate students: http://e-catalog.jhu.edu/grad-students/graduate-specific-policies/#misconduct

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 .