Syllabus

CSCI 136: Fall 2018
Handout 1
7 September

Data Structures and Advanced Programming

Instructor (11a.m.)	Prof. Bill Lenhart (Lectures & Labs)
Office	TPL 304
Phone	597-2371
Email	wlenhart@williams.edu
Instructor (10a.m.)	Prof. Bill Jannen (Labs)
Office	TCL 306
Phone	597-4509
Email	jannen@cs.williams.edu
Lectures	MWF 9:00-9:50am (Lenhart) in Schow 030B
Labs	W 12–2pm, 2–4pm (Due Sundays at 11pm)
Web Page	`http://www.cs.williams.edu/~cs136`

Texts

We will be using the $\sqrt{7}$ edition the following text book:

- *Java Structures: Data Structures in Java for the Principled Programmer,* $\sqrt{7}$ *E*dition, Duane Bailey.

Do not use earlier editions! A PDF version is available on the course website. We have also printed copies of the text book as a course reader. We encourage you to take a copy of the course reader; your term bill will be charged whether you take a copy or not, but we will reuse unclaimed books for future courses.

Course Objectives

This course couples work on program design, analysis, and verificationfina with an introduction to the study of data structures. Data structures capture common ways in which to store and manipulate data, and they are important in the construction of sophisticated computer programs. We will use the Java programming language in class and for the assignments.

Students will be expected to write several programs, ranging from the short and simple to the more complex and challenging as the semester progressess. Since one of our goals in this course is to help you learn how to write large, reliable programs composed from reusable pieces, we will be emphasizing the development of clear, modular programs that are easy to read, debug, verify, analyze, and modify.

We will use the computers in TCL 216 & 217a for the programming assignments. You will be given door codes to access this room once the semester begins.

Course Work

Workload The work that you should expect to engage with, beyond the scheduled lectures and weekly lab meetings, will involve

- Reading the text: 12-15 pages, on average, per lecture
- Preparing for the weekly programming labs
- Completing the weekly labs
- Completing a modest number of problem sets
- Studying for the mid-term and final exam

Some students program quickly but read slowly, some do the opposite. The typical student should expect to spend at least 10 hours a week beyond the scheduled lecture and lab hours. If you find yourself spending substantially more time than that on a regular basis, discuss the issue with your instructor.

Labs. On most weeks, there will be lab programming assignments. **Attendance in lab is mandatory**: there are valid reasons to miss lab, but any unexcused lab absence will result in course failure.

All programs will be graded on design, documentation and style, correctness, and efficiency. Programs should be turned in electronically by 11pm on the due date, typically the Sunday following lab. Each student may use a maximum of **three free late days** during the course of the semester. A late day permits you to hand in an assignment up to 24 hours late, without penalty. Once late days are exhausted, late work will not be accepted; the most recent on-time submission will, however, be graded so that some partial credit may still be obtained. Using a late day requires that you

1. Email your instructor in advance,
2. By the regular submission deadline, submit the work you have completed so far, with a comment at the top of your README.md file that says: “Incomplete assignment, late day requested.”
3. When you have submitted the completed assignment, email your instructor and remove the “Incomplete assignment” comment from the README.md file.

Late days are provided to help students deal with unforeseen circumstances and to allow some balancing of occasional uneven work demands. They should be used judiciously; if you find yourself struggling with the workload of this (or any other) course, we endourage you to reach out to your instructor.

Exams. There will be one midterm and one final exam. The midterm will be scheduled during your lab period on Wednesday, October 17, and it will replace the lab for that week. The final exam will be self-scheduled.

Grading. Grades will be determined as follows:

Final exam:	30%
Midterm exam:	25%
Programs/Labs:	35%
Problem sets:	5%
Engagement:	5%

Engagement is a subjective evaluation of how you interact with the material and the class. We will use the full range from 0 (e.g., submits required work, attends some classes) to 5 (e.g., attends all sessions, participates in class, refers to the readings, raises new topics, shares knowledge with other students). Learning is collaborative; this is our way to reward students who positively affect the experience of fellow class members.

Honor Code

The Computer Science Honor Code can be found here. Since interpretation of how it might apply in individual courses might vary, we provide additional detail here.

The work for this course spans a variety of types that differ with respect to how the honor code applies. The types of assignments given in this course are program design, program implementation, thought questions, problem sets, and exams. Below we describe how the honor code applies to each type of assignment, including examples of permitted and prohibited behaviors. These examples are not exhaustive; if you have any questions about how the honor code might apply in a particular circumstance, please discuss it with your instructor.

Single-Author Programming Assignments The successful completion of a lab assignment involves broadly three steps: Program design, program implementation, and responses to ”thought questions”, which typically ask you to reflect on some aspects of the lab. Each individual is responsible for producing their own work. Examples of permitted and prohibited activities for single-author labs are described here.

Program Design A number of labs require that each student prepare a *design document* which is a brief, high-level implementation plan that typically describes the intended data structures, code organization, and order of implementation. Students in the course are permitted to participate in discussions with one another about program design, but should ultimately produce their own written design document. Any collaboration on program design should be explicitly noted in the design document. Students are not permitted to share their design documents or detailed descriptions of their contents with other students nor are they permitted to uses resources beyond those provided by the instructor in the creation of program designs.

Program Implementation (code) The programs written by student should represent their own work. Students are permitted to ask other students in the class questions of clarification, language syntax, and error message interpretation, but are not permitted to view/share each others code or design documents, Further, students should not use any resources beyond those directly provided by their instructor (so-called *outside sources*; in particular, viewing descriptions of solutions similar problems from outside sources is prohibited.

Thought Questions Most labs include a small number of what are referred to as thought questions. Thought question responses should be the work of the individual students; students should not discuss thought questions with anyone other than their instructor and TAs.

Group Programming Assignments For some of the labs, students will be offered the option of working with a partner. Both a partner pair and a single student working alone are referred to as a *group*. Each group submits a single design document, a single set of thought question solutions, and members of the same group may discuss any aspect of the assignment with one another. Interactions between groups are subject to the constraints described above under **Single-Author Programming Assignments**.

Problem Sets Students may collaborate with other students in the course on developing approaches for solving problem set problems as long as such collaboration is explicitly noted in the submitted work. Students are not permitted to share/view the written work or detailed notes of others or to use any outside resources in the development and writing of problem set solutions.

Exams The mid-term exam and final exam will both be scheduled exams. They are closed-book exams. No resources may be accessed while taking the exams with the sole exception of asking the instructor questions of clarification.

Help!!!

Help. We all need it. There are many resources available when *you* need it. You are encouraged to discuss any questions, concerns, difficulties, or thoughts about the course with your instructor (or anyone else named Bill who happens to be teaching the course). In addition, TAs are available to help you with challenges you might face as you work through the course material and lab assignments. If you find yourself facing challenges beyond the typical, we encourage you to reach out. Talk to your instructor, a friendly face from the Dean's Office, or some of the many professionals across campus who stand ready to help, including

- The Peer Tutor Program: Tutors can be arranged when 1-1 help is required beyond that available from your instructor and TAs. `https://academic-resources.williams.edu/peer-tutor-program/`
- Math & Science Resource Center: Support is available for students grappling with the more quantitative aspects of their coursework. `https://academic-resources.williams.edu/math-science/`
- Accessible Education and Disability Support Center: Students with documented disabilities may require accommodations in certain situations. If that's you, take advantage of the options available.

  `https://academic-resources.williams.edu/disabilities/`
- The Health Center: Sometimes your challenges are not course-related. The Health Center provides a range of medical, psychological, and health/wellness services. `https://health.williams.edu`