C Sc 227 Program Design and Development

University of Arizona, Spring 2011, Syllabus

University Catalog Description (4 units) Programming in an object-oriented language such as Java for students with significant prior programming experience. Topics include types, control structures, classes, arrays, sorting, algorithm analysis, recursion, program development, and implementation of abstract data types using linear structures and binary trees.

Prerequisites This course is for students with significant programming background who can learn a new programming language quickly. Many of the same concepts you learned in another language will be "covered" on an as-needed basis: control structures (if, for, while), functions and parameters, and arrays. Many students take this class after ECE 175 or C Sc 127A.This course is an accelerated one-semester alternative to the two-semester C Sc 127A and C Sc 127B sequence. It has two semesters of concepts in one semester.

Instructor Rick Mercer http://www.cs.arizona.edu/people/mercer. Office 727 Gould Simpson ~~Mondays and Wednesdays 11:00-12:00 noon,~~ Tuesdays 13:00-14:00, Fridays: 11:30-12:30, and by appointment 520.621.6126 mercer@cs.arizona.edu

When/Where Common Lecture: MWF 10:00-10:50 in 202 AME. Recitation sections: Various times/locations

Web Site The course website has all current course information. It is updated almost daily with assignments, lecture outlines, and programming projects: https://sites.google.com/site/cs227s11/

Textbook The text book was written by your instructor. It is provided free as pdfs or as a perfect bound printed book from Lulu for 13.29 plus shipping: http://www.lulu.com/product/paperback/c-sc-227-program-design-and-development/14449789

Course Philosophy and Learning Objectives The instructor and section leaders will facilitate lectures and recitation sections to help you learn the process of programming and guide you in developing skills that will serve you in this course, future courses, and industry. We ask you to take responsibility for your learning by coming to class every day, preparing for class, participating thoughtfully in all daily class activities, and respecting others. When you successfully complete this course, you will be able to

- Analyze problems, design and implement computer based solutions
- Understand how to use existing types and build new types (using Java classes)
- Design experiments to validate code using an industry-level testing framework:

  "Test-driven development was the most important thing I learned in college", former student now working as a software developer.

  "When I told them I was doing unit testing, they made me the offer immediately", former student now working in industry.

- Debug existing code using an industry-level debugging tool
- Understand and apply encapsulation
- Use Java generics to implement generic collections
- Implement abstract data types using linked structures

- Know classic algorithms to search, sort, insert, and remove elements in collections
- Implement recursive algorithms
- Understand and implement hierarchical data structures, specifically binary trees.

Course Grade	Letter Grades
40% Programming Projects	A >= 90%
10% Assignments	B 80.0 through 89.9
15% Test 1 Friday 18-February	C 70.0 through 79.9
15% Test 2 Friday 1-April	D 60.0 through 69.9
20% Final Friday 6-May 10:30	E < 60.0

Test Makeup If you are unable to take a test at the scheduled time for a valid reason, you must contact Rick to get permission to take a makeup test. Unless you are physically unable, call Rick's office (621-6126) or send an email BEFORE the test with as much lead time as possible. Describe how you can be contacted to discuss how to make up the test. Without PREVIOUS notification and a valid excuse (documentation may be required) you may not be able to make up the missed test.

Programming Projects Each individual programming project will have its own specification and due date. All programming projects are worth100 points. If you do not turn in a project on time, you will receive an automatic 10 point reduction for every 24 hours or portion thereof. A project that is 24 hours and 1 minute late would result in a loss of 20 points. Projects will no longer be accepted after 5 days from the due date for a maximum score of 50 points. You will receive a 0 if nothing is turned in within 5 days of the due date. Your final score cannot be less than 0. Recommendation: Begin your programming projects as soon as possible.

Assignments This portion of your grade consists of homework, small programs, and JavaBats. Since we have one programming project almost every week, assignments must be completed during weeks when you also have a project.

Absence Policy You are encouraged to attend all lectures and sections. Some lectures will have graded in-class activities that count for the assignments portion of the grade. If you miss that lecture, you will be able to turn them in later as prescribed on the assignment. It is your responsibility to check the Assignments page of the course web page.

Students with Disabilities If you anticipate barriers related to the format or requirements of this course, please meet with me so that we can discuss ways to ensure your full participation in the course. If you determine that disability-related accommodations are necessary, please register with Disability Resources (621-3268; drc.arizona.edu) and notify me of your eligibility for reasonable accommodations. We can then plan how best to coordinate your accommodations.

Software at Home We will be using Java 6 and Eclipse 3.6 (Helios) in the lab and in lecture. Both are free and available on Windows, Linux, and Mac OS X. You can set up your home computer with the same tools used in lecture and in lab.

Classroom Behavior The University of Arizona has an explicit policy on disruptive behavior: http://web.arizona.edu/~policy/disrupt.shtml Included in disruptive behavior are cell phone, PDA, and pager use, talking during lectures, sleeping, eating, arriving late or departing early (without prior notification), and newspaper reading. Such activities distract others and interfere with instructional activities. Again, students should use class time to further their learning, through active engagement with

the material. Please treat each other with respect. Laptops are allowed, but if other students report them as disruptive, laptop use may be banned also.

Academic Integrity and Penalties Programming projects and assignments in this course require individual attention and effort to be of any benefit. Unless otherwise specified in the published assignment, all work is expected to be that of each student alone. You may not consult with others, except in ways specifically authorized by the course instructor. Students are responsible for understanding and complying with the University's Code of Academic Integrity. The Code can be found at this link http://deanofstudents.arizona.edu/codeofacademicintegrity. The full text is also available from the Office of the Dean of Students in Room 203 Old Main.

Among other provisions, the Code demands that the work you submit is your own, and that graded programs and exams will not subsequently be tampered with. The Code also demands that you do not copy code when it is part of a published class assignment. It is immaterial whether the copying is done electronically, by retyping the code, looking at another's computer screen, or any other means. Violations of Academic Integrity will result in a report filed to the Dean of Students. Sanctions include receiving an E for the course, even if it is a first violation. If other reports have been filed from any department, the Dean of Students may issue more severe sanctions including suspension or expulsion from the university.

You are better off receiving 0 for one project rather than an E for the course and a report on your University record. Avoid Sanctions by beginning your projects as soon as possible. Do not wait until the due date! Do not look at another person's test while the test is in progress. Do not copy files. Do not give your code to anyone even if the other person promises not to turn it in as their own, in which case you who did all the work may suffer the same sanctions as the cheater. The automated grading tool we use encourages multiple submissions that provide feedback to allow 100% scores if you put in the effort.

Section Leaders This course has a weekly scheduled lab facilitated by your section leader. These labs have a variety of activities including a chance to review lecture material, ask questions in a small classroom setting, and develop algorithms and code as a whole or in teams. All section leaders have taken CSC courses with section leaders. They also participate in a credit course designed to learn how to be an effective section leader. Section leaders grade programming assignments and tests, develop and edit course materials, help make pedagogical decisions, keep the instructor up to-date with student progress, and provide assistance in the lab. You can find your section leader's name and email at https://sites.google.com/site/cs227s11/teachers

Subject to Change Information contained in this course syllabus other than the grading and absence policy is subject to change with reasonable notice.