Introduction to Programming II

Computer Science 162 - Spring 2018 - Ms. Katz
Tuesdays 8 AM - 9:50 AM in Roddy 147 (sometimes in Linux Lab) and Thursdays 8 AM - 9:50 AM in Roddy 136

Instructor
- Ms. Beth Katz - Beth.Katz@millersville.edu - http://cs.millersville.edu/~ekatz
- Office: Roddy 139
- Office Hours: 10:30 AM to 11:30 AM Monday, 10 AM to noon Tuesday and Thursday, and by appointment
- Phone: 871-4313 (office on class days; Email is most effective way to contact me)
- Class Web Page: http://cs.millersville.edu/~ekatz/cs162.html

Prerequisites
You are expected to have earned a C or better in CS 161 before taking this course.

Required Textbook
Data Structures and Other Objects using Java (fourth edition), Michael Main, Addison Wesley. ISBN 9780132576246

Goals
Our main goals this term are to further develop your problem solving and software engineering skills (particularly encapsulation), to explore and implement some classic data structures, and to examine various classic algorithms for solving particular types of problems. You will be asked to write many programs. These assignments are a very important part of the course. The assignments build on each other and reinforce concepts from lecture. Don't fall behind on the reading or programming assignments.

Grading
Test 1 (March 1) = 25%
Test 2 (April 12) = 25%
Final exam (Thursday, May 10, 10:15am -12:15pm) = 25%
Assignments (earned points / total points available points of projects; assignments will have varying point values) = 25%

I use a ten-point grading scale (93 = A; 90 = A-; 87 = B+; 83 = B; ...). You must attempt all examinations and assignments to pass the course. You must complete the programming and laboratory assignments by yourself unless explicitly told otherwise. Pair programming, as allowed on specific assignments as noted, is the only collaboration allowed. I report academic dishonesty to the provost.

Attendance
I expect you to attend both lecture and lab, read the text as noted, and participate in class discussions. I will take roll. Getting notes from a friend is not sufficient. Tests emphasize what we have covered in class. Missing class reduces your understanding of the material and will make passing the course more difficult. Missing the hands-on lab experiences will make the assignments substantially more difficult. Come to class prepared to learn.

Academic Honesty
You may ask me questions at any time. Copying or any collaboration with anyone else on tests is not permitted and may result in failure in the course.

You will be allowed to use pair programming for some assignments. You must collaborate closely with your partner. However, any other copying or collaboration with other people is forbidden. Refer to the university policy on academic honesty for details and penalties.

You may discuss approaches to solving the problem as well as general aspects of the technique we are studying. You may also seek aid in resolving compiler messages. However, if you can't explain why you did something, you probably didn't do it yourself. Finding or purchasing a solution on the internet or elsewhere and turning that in as your own work is also plagiarism. When in doubt, ask me.
Tests
Tests will be in the classroom. They will be cumulative with closed book, notes, and neighbor. The only allowed tools are pen or pencil, blank paper, and your brain. If you miss a test, you will receive a grade of zero for that test. Exceptions may be made at my discretion for reasons of illness or university excused absences. Contact me immediately, preferably before the exam, if you have a problem.

Programs - Early, Late, and Faulty
There are no late programs. A program must be delivered by the time and date specified for the assignment to my katz162 account. If your program is not complete, submit what you have for partial credit. Develop your programs incrementally, and you will always have something to turn in for partial credit. I do give partial credit, and some points are better than zero points.

Each assignment will note how many points it is worth. Some assignments are worth more than others, but you should attempt all of them. The number of points you accumulate on assignments will be summed and that total will be divided by the total points available.

All programs must compile properly using Java in Eclipse on the MU CS department's Linux computers without compiler errors. Programs that do not compile have a maximum grade of 25% of the assignment's points.

A program that compiles cleanly is not necessarily correct. I will deduct points for missing, incomplete, and incorrect features. Be sure your program meets the problem specification. Read your programs for errors and test them with several suitable test sets. Note that I am extremely good at breaking programs. You should try to break your program before I do.

I also read your programs. I will deduct points for poorly indented and commented programs as well as those that use poor programming practices or do not use the techniques and data structures specified in the assignment. 'A' does not mean adequate. Refer to example programs from class as models.

Contacting Me
If you have a question, don't wait to get help. Ask questions in class. Work the example problems and warm-up exercises. This is a cumulative subject, and it doesn't get easier. Practice helps.

Ideally you will ask me questions in person. I have office hours as noted above and by appointment.

I am also accessible through email. I check my mail several times a day. If you have a question about your program, submit the source code as if you are turning it in and then send me email asking your question and letting me know you submitted the program. If a question or answer pertains to many people, I will send a general answer to the entire class by email.

Expectations
This is a 100-level course, but it is definitely not easy. It is substantially more difficult than CS 161. Becoming a good computer scientist takes practice. I expect you to attend class regularly, read the text, do the warm-up exercises, and participate in class. Written class notes copied from a classmate do not capture the process of developing the problem solutions. Don't fall behind on the programming assignments. This is a cumulative course in a technical subject that has its own vocabulary. Pay attention, schedule enough time, and get help early.

On some (not all) Tuesdays we will meet in the Linux lab for hands-on practice. I will give you the handout describing the lab activity at a class before the lab. I expect you to read through it before you come to lab and sometimes do some pre-work to get ready for lab. This will make the lab period much less stressful. You are expected to remain in the lab for the full lab period or until you have successfully completed all work. Most assignments will require further work outside the lab period. Expect to spend considerable time on the programming assignments outside class.

This is a computer science course. I expect you to be comfortable using the web, email, and Eclipse. I expect you to check your university email regularly and at least once a day. Examples, answers to questions other students have asked or problems I've seen, and other helpful information will be available through email, files in my examples folder, or on the course web page. Take advantage of these resources.
Submitting and Grading Assignments
You will turn in all labs and assignments using an automated assignment management system. Be logged into
the Linux system and using a terminal. Eclipse will have stored each project in a separate directory. Move to
the directory where the source files for this project are stored. Use an `ls` command to see that your Java
source code files end in `.java`. For some assignments, some of your source code may be in subdirectories. Use
the `submit` command, choose `katz162` (my account for grading), and then choose the assignment name. Your
project files are copied to the appropriate place in my `katz162` files with your account name noted. If you
decide you have a better version and the due date hasn't passed, you may submit again. The old version in my
files is renamed, and the file you indicate is copied to my file space. Unless there is a problem, I do not look
at these older versions. There is no penalty for submitting a reasonable number of versions. You may use the
`submitlog` command to check what you have submitted and when.

When I grade the assignment, your most recently submitted project source code runs through the compiler.
The executable programs are run with several sets of test data, and the results are recorded. I do not
necessarily type in data to each program each time I grade an assignment although I will do some interaction
with graphical versions. It is your responsibility to make sure that your program reads the data in the correct
order and format. Don’t expect extra input. Don’t change the input requirements to suit your program. Unless
otherwise specified in the assignment, all data should be read from standard input just as if it had been typed
at the keyboard.

I personally read and evaluate your final submission for each assignment. I check whether it gives the correct
answers for each of several test sets. Does it print the output in the correct format? What happens when your
data structure is empty, when elements are added to or deleted from the beginning, end, or middle, or when
it is full? Does it use the programming concepts we’re studying? Did you pass parameters appropriately? I
examine the program’s style including comments and indentation. Do the functions each do one thing well?
Just getting the correct answers is not enough to earn full credit.

I also notice unusual ways of approaching problems, spelling mistakes, and similarities between programs. I
am very annoyed when I spend time grading a program and run across it a second time when grading a
program submitted by another student. Don’t copy another student’s program; it’s cheating.

I record comments about your program as I read it. My comments state how many points I am counting off
and why. For example

\(-10\) Your total variable is not initialized to zero.

All my comments to you are gathered into an email message containing your grade for the assignment. The
email is sent to your account millersville email.

Using Available Resources
Read the textbook. But read it not as a novel or something to be memorized. Read it in small doses when you
can concentrate. Work the Self-Test Exercises. There are solutions at the end of each chapter. Skim chapters
before we cover them in class. Read sections again after we have talked about them in class.

The course web page at `http://cs.millersville.edu/~ekatz/cs162/` has information about the course. The
eamples developed in class are linked from there or are in `/home/grader/katz162/examples` on cs. The
textbook programs are on the authors’ web site; there’s a link on the course page.

Have Fun
I love being a computer scientist. I can create something out of my imagination and share it with others.
Programs are often like puzzles. This is fun stuff, and I hope you’ll enjoy it. If an assignment doesn’t seem
fun, try to think of it as a puzzle. Put together the frame and then work on small parts.

Special Needs
Anyone with special needs should contact me as soon as possible.

Statement on Title IX Responsibilities for Faculty
Millersville University and its faculty are committed to assuring a safe and productive educational
environment for all students. In order to meet this commitment, comply with Title IX of the Education
Amendments of 1972, 20 U.S.C. 1681, et seq., and act in accordance with guidance from the Office for Civil
Rights, the University requires faculty members to report to the University’s Title IX Coordinator incidents of
sexual violence shared by students. The only exceptions to the faculty member's reporting obligation are
when incidents of sexual violence are communicated by a student during a classroom discussion, in a writing assignment for a class, or as part of a University-approved research project. Faculty members are obligated to report to the person designated in the University Protection of Minors policy incidents of sexual violence or any other abuse of a student who was, or is, a child (a person under 18 years of age) when the abuse allegedly occurred.

Information regarding the reporting of sexual violence, and the resources that are available to victims of sexual violence, is available at http://www.millersville.edu/socialeq/title-ix-sexual-misconduct/index.php