Required Text:


I have not requested that the UR book store order this book - you will save about $15 by purchasing it online.

Course Description:

Think for a moment about your parents’ generation:

Cameras required film; recorded music was played using vinyl or cassettes; renting a movie required a video cassette player; and calling while away from home often required locating a pay phone.

Think now about your own generation:

you use digital cameras, CDs, DVDs, mobile phones. Digital music and video for iTunes is ubiquitous.

The transition from your parents’ analog generation to your own digital generation has been swift, revolutionary, challenging and in some ways deeply mysterious. How, for example, can a sequence of 0s and 1s accurately represent the art of our favorite musicians? How can a single hand-held device allow us to communicate by voice, picture, video, and text, while simultaneously serving as a Web browser, email client, personal digital assistant, music player, and game console, while a mere 20 years ago, some of those services didn’t even exist, and the remaining ones would have required multiple large and expensive devices? The fundamental principles that facilitate the current generation of digital communication, information processing, and computation provide the motivation for this course.

The purpose of this course is to provide an introduction to computer science as a discipline. Computer science is much more than programming. We like to say that programming is to computer science as microscopy is to biology. No one becomes a competent biologist without mastering the skills necessary for microscopy; but being a master at microscopy does not a biologist make, nor does a biologist spend all of her time behind the microscope. A similar argument holds for computer science and programming:

Learning to master the basics of computer programming is necessary to become a computer scientist. However, the discipline of computer science is about discovering accurate solutions to interesting, challenging problems that have real-world impact, and about representing these solutions in a precise way so that computers can assist with solving the problems.

Therefore, in this course we will investigate problem solving using computer programming. All digital communication, information processing, and computation systems rely on precise algorithms expressed as computer programs. We will study the capabilities and limitations of such algorithms by considering abstract representations of the complex processes required by some commonly-used digital systems, e.g.:

- mobile applications;
- digital image representation and processing; and
- simple computer network servers.

You will also learn the basics of computer programming using the Java programming language, covering topics such as object-oriented programming, flow control structures, string processing, simple data structures, and recursion. Your programming projects for the course will require you to implement in Java working solutions for systems such as those discussed above.
Although the programming work in this course will focus on the application of programming to digital communication and information processing systems, the programming skills you will develop along the way will be applicable to a wide variety of areas. You will also have a much better appreciation for the discipline of computer science — that it is much more (and much more interesting) than just programming.

This course satisfies the Symbolic Reasoning Field of Study (FSSR), of which a description is available on the Office of the Registrar's page:

http://oncampus.richmond.edu/academics/registrar/grad/gened.html

Labs and Homework:

- Lab sessions will be held each week of the semester. A description of each week's lab will be handed out in advance, and you will be expected to plan your approach to the assignment before the start of that lab session.
- Lab assignments will be submitted electronically (via Netfiles), and will be accepted until 11:00 PM on the Monday following your lab session, unless a different due date is listed in the assignment. No lab assignments will be accepted after that time, but your lowest score on submitted lab assignments will be dropped. (Hence, it is in your best interests to submit all assignments on time, even if incomplete.)
- There will be some homework assignments, roughly weekly, though we will skip some weeks. Assigned homework will be collected and graded. Late homework will not be accepted. No homework grades will be dropped.
- We will use the course Web page and email for assignment-related information. It is your responsibility to check both frequently.

Exams:

- Two in-class written exams, limited to the 50-minute class block, and an in-class comprehensive final exam, limited to the 3-hour exam block. It is my policy not to give make-up exams. If you are absent on the day of an exam, your final exam grade will be substituted for the grade of the missed test. I have found that this is less disruptive and stressful for both you and me.
- A take-home programming exam, to be assigned and completed during the final weeks of the semester. This programming exam may be submitted late, subject to a penalty of 10% per calendar day.

Grading Policy:

Final letter grades will be assigned per the traditional 10-point scale (≥ 90% is at least an A–; < 90% but ≥ 80% is at least a B–; etc.), according to the following percentages:

- Lab Assignments/Homework 30%
- In-Class Written Exams 25%
- Take-Home Programming Exam 25%
- Final Exam 20%

Attendance Policy:

- You are expected to attend each class period and each lab session for its duration. If you must miss a class or lab, you are responsible for any associated material. If there is a class or lab that you must miss, please inform me in advance.
- Any student with an excessive number of absences is subject to a failing grade of V.

Honor Code:

- Unless provided to you by me, you are not permitted to view or use existing assignments, tests, or solutions in any form, whether they be from a previous offering of this course or Internet-available.
- In-class written exams must be completed by you without assistance from any other person or source.
- The take-home programming exam must be treated as a take-home, open-book exam. You may use only your textbook, class notes, and any other instructor-approved source. You may not consult anyone other than the instructor, who encourages asking questions but reserves the right not to answer (just as you would expect with any exam).
- Lab assignments and homework assignments may be discussed with others, but are subject to the empty hands policy:
You may freely discuss ideas with other students, but each student must leave the discussion without any written or otherwise recorded material.

You may not work directly with any other student on the completion of the assignment. Any manifestation of copying another student’s work for your own (whether digital, hand-written, oral, etc.) or working together on an assignment is not permitted.

- Failure to comply with these policies will be treated as an Honor Code violation.

Special Notes:

If you have a disability and want to discuss appropriate accommodations, or if your observance of a religious holiday presents a conflict, please contact me as soon as possible.

Course Outline:

Following is a basic list of topics to be covered this semester. Additional topics may be added as necessary. The order of these topics is subject to change.

- Being Digital
- Encoding in Binary
- Programs, classes, and methods
- Android application structure
- Providing a GUI Interface
- Conditional Execution
- Processing Strings in Java
- Iteration
- Recursive Class Definitions
- Indexing and Image Manipulation
- Arrays and Image Manipulation
- Image Compression, Color Quantization

Important Dates:

Regular exam dates are subject to change based on the pace of the course.

Exam 1: Friday 17 February
Exam 2: Friday 30 March
Final Exam: S2: Friday 27 April 2:00 pm – 5:00 pm
Spring Break: Saturday–Sunday 3–11 March