CS2133: Computer Science II
Spring 2020 Syllabus

Prof. Christopher Crick

Abstract

Computer Science II is the follow-on course to Computer Science I, which introduced you to programming and the Java language. This class will deepen and expand your programming repertoire, and by the end you should be competent in designing, implementing and debugging medium-sized Java programs with graphical user interfaces. In addition, this class will introduce you to the science aspect of computer science, providing you with tools to analyze your programs and predict their behavior and efficiency ahead of time.

Professor

- Dr. Christopher Crick
- Office: MSCS 213
- Lab: MSCS 214
- Office hours: T 10-11:30 and 1-7, or by appointment, or walk-in
- Email: chriscrick@cs.okstate.edu

TA

- Sayanti Roy
  - Office: MSCS 214
  - Office hours: W 10-12
  - Email: sayanti.roy@okstate.edu
- SM al Mahi
  - Office: MSCS 202
  - Office hours: M 3:30-5:30
  - Email: smahi@okstate.edu

Course Meetings

- MWF 2:30-3:20, Agricultural Hall 108

Text

- Walter Savitch, *Java: An Introduction to Problem Solving and Programming* (8th Edition)
- This, or any other Java programming reference, is **recommended, but not required**.

Grading

- Assignments: 60%
- Project Euler programming exercises: 10%
- Exams: 30%
- Extra credit available by completing additional programming exercises
- Total grade will be modified by attendance

Grade Breakdown

- A: 90%
- B: 80%
- C: 70%
- D: 60%
- We reserve the right to curve these percentages downwards if necessary, but they will not be curved upwards. If you score 90.0%, you will earn an A.

Topics Covered

- Programming
  - Review of Java I
  - Packages
  - The class model
  - Interfaces, abstract classes, final classes, wrapper classes
  - Constructors
  - Static methods and variables
  - Arrays
  - Java data structures
  - Inheritance and polymorphism
  - Graphical user interfaces (GUIs)
  - File, command line and GUI I/O
  - Exception handling
  - Generics
  - Javadoc
  - Testing and debugging
- Theory
  - Information hiding and encapsulation
  - Recursion
  - Running time analysis and Big-O notation
  - Sorting and searching
  - Stacks, queues, linked lists
  - Hashtables and trees

Policies

- Assignments will ordinarily be due on Wednesdays at noon. We will often go over assignments in class the next Monday, which means that any assignment turned in more than 5 days late will receive no credit. Late assignments will be penalized 10%. If a legitimate academic or medical conflict comes up, this penalty can be waived, but any such extension must be negotiated *in advance*.
- Assignments should be handed in to the D2L dropbox (online.okstate.edu) as a single .zip file. Programs must compile and run simply from the command line, or they will not receive full credit.
- There will be two exams, a midterm and a comprehensive final. These will account for 30% of your grade, and the final counts for twice as much as the midterm. You will be permitted one sheet of handwritten notes for each.
- Attendance at lecture is required. An attendance sheet will be circulated during randomly selected lectures. Missing two such lectures will not affect your grade, but every additional missed attendance will penalize your final grade by one whole percentage point. Within reason, requests for excused absences will be accepted, as long as they are made in advance.
- Academic integrity is taken very seriously. You are permitted (and indeed encouraged) to discuss the course material with fellow students in general terms, but the programs you write must be your own. **Code copied from each other or found on the net will result in an automatic zero for the assignment**, and depending on the egregiousness of the offence may result in earning an 'F!' for the course and facing academic disciplinary measures.
- That said, you are welcome to copy code from your own previous assignments, from programming snippets that we go over in lecture, or from the textbook.
- Project Euler (http://projecteuler.net) is a web site that contains several hundred progressive mathematical problems that can be solved by writing a small program. Many of the problems require clever solutions that think about run-time analysis; your solutions should execute in about twenty seconds or less. You are required to solve ten of them of your choice over the course of the semester. In addition, you may solve up to ten more of them for extra credit, to make up for poor test or homework scores. Each successfully completed extra credit problem will raise your grade one whole percentage point. **Note:** Problem 1 on the Project Euler site will be demonstrated in class. You may not submit that problem for credit.

Class schedule

- January 13 (M): Course introduction
- January 15 (W): Remember this?
- January 17 (F): Object-oriented programming. Intro slide due.
- January 20 (M): Martin Luther King Day (**No class.**)
- January 22 (W): Commenting and refactoring.
- January 24 (F): Make up, catch up, or no class.
- January 27 (M): Testing and debugging.
- January 29 (W): Recursion. Assignment 1 due.
- January 31 1 (F): Inheritance.
- February 2 (M): Polymorphism.

- February 4 (W): Interfaces.
- February 6 (F): Graphical User Interfaces (GUIs).
- February 10 (M): More GUIs.
- February 12 (W): The event loop. Assignment 2 due.
- February 14 (F): Widgets.
- February 17 (M): Model-view-controller architecture.
- February 19 (W): Timers and animation.
- February 21 (F): Exceptions.
- February 24 (M): Streams.
- February 26 (W): **No class**. Assignment 3 due.
- February 28 (F): **No class**. Five Euler problems due.
- March 2 (M): More stream I/O.
- March 4 (W): Objects and sockets.
- March 6 (F): Threads.
- March 9 (M): Midterm review.
- March 11 (W): Midterm.
- March 13 (F): Make up, catch up, or no class.
- March 16 (M): Spring break. **No class**.
- March 18 (W): Spring break. **No class**.
- March 20 (F): Spring break. **No class**.
- March 23 (M): Generics.
- March 25 (W): Orders of growth. Assignment 4 due.
- March 27 (F): Sorting. Five more Euler problems due.
- March 30 (M): Collections.
- April 1 (W): Linked lists.
- April 3 (F): Make up, catch up, or no class.
- April 6 (M): Hashes and heaps.
- April 8 (W): Priority queues. Assignment 5 due.
- April 10 (F): Red-black trees.
- April 13 (M): Bits and bytes.
- April 15 (W): Dynamic programming.
- April 17 (F): Make up, catch up, or no class.
- April 20 (M): Under the hood.
- April 22 (W): Lambda expressions. Assignment 6 due.
- April 24 (F): Regular expressions.
- April 27 (M): Final exam review.
- April 29 (W): Final thoughts.
- May 1 (F): Make up, catch up, or no class. Extra credit Euler problems due (up to ten).
- May 8: (F) Final exam (2 pm)