

CS2133: Computer Science II

Fall 2019 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 320

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 Canvas dropbox (canvas.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. 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

- August 19 (M): Course introduction
- August 21 (W): Remember this?
- August 23 (F): Object-oriented programming. Intro slide due.
- August 26 (M): Commenting and refactoring.
- August 28 (W): Testing and debugging.
- August 30 (F): Make-up, catch-up, or no class.
- September 2 (M): Labor day. **No class.**
- September 4 (W): Recursion. Assignment 1 due.
- September 6 (F): Inheritance.
- September 9 (M): Polymorphism.
- September 11 (W): Interfaces.

- September 12 (F): Make-up, catch-up, or no class.
- September 16 (M): Graphical User Interfaces (GUIs).
- September 18 (W): More GUIs. Assignment 2 due.
- September 20 (F): The event loop.
- September 23 (M): Widgets.
- September 25 (W): Model-view-controller architecture.
- September 27 (F): Make-up, catch-up, or no class.
- September 30 (M): Timers and animation.
- October 2 (W): Exceptions. Assignment 3 due.
- October 4 (F): Streams. Five Euler problems due.
- October 7 (M): More stream I/O.
- October 9 (W): Objects and sockets.
- October 11 (F): Make-up, catch-up, or no class.
- October 14 (M): Midterm review.
- October 16 (W): Midterm.
- October 18 (F): Threads.
- October 21 (M): Generics.
- October 23 (W): Orders of growth. Assignment 4 due.
- October 25 (F): Fall break. **No class.**
- October 28 (M): **No class.**
- October 30 (W): Sorting.
- November 1 (F): Collections. Five more Euler problems due.
- November 4 (M): Linked lists.
- November 6 (W): Hashes and heaps. Assignment 5 due.
- November 8 (F): Priority queues.
- November 11 (M): Red-black trees.
- November 13 (W): Bits and bytes.
- November 15 (F): Make-up, catch-up, or no class.
- November 18 (M): Dynamic programming.
- November 20 (W): Under the hood. Assignment 6 due.
- November 22 (F): Lambda expressions.
- November 25 (M): Regular expressions.
- November 27 (W): Thanksgiving Break. **No class.**
- November 29 (F): Thanksgiving Break. **No class.**
- December 2 (M): Final exam review.
- December 4 (W): Final thoughts.
- December 6 (F): Make-up, catch-up, or no class. Extra credit Euler problems due (up to ten).
- December 11 (W): Final exam (2 pm)