Abstract

Computer Science I will introduce you to the craft of programming and to the Java language. By the end of the course, you will be proficient at translating problems into syntax that can be interpreted and executed by a computer. You won’t yet have the skills necessary to create anything you can imagine — that will have to wait for Computer Science II — but you will have built an excellent foundation.

Professor

- Dr. Christopher Crick
- Office: MSCS 213
- Lab: MSCS 214
- Office hours: Tuesdays 3-7, via the text “Chat” function on the course’s Canvas page. If necessary, we will proceed from there to a videoconference. You should also feel free to e-mail any time with questions, and videoconferences can be scheduled outside of office hours as well. The professor will make every effort to respond to e-mails within 24 hours.
- Email: chriscrick@cs.okstate.edu

Objectives

- Use variables, control structures, arrays and method definitions to produce useful text-based programs which solve both toy and real-world problems.
- Design well-structured, encapsulated, self-documenting code which can be maintained, updated, and improved as the course progresses.
- Demonstrate robust code which responds gracefully to errors and unexpected user behavior.
- Choose appropriate data structures and data types for representing problems, explain the rationale for such decisions, and express the underlying computational and memory processes which pertain to the various choices.

Course Meetings

No meetings for the online course. Lecture videos will be posted to canvas.okstate.edu.
Text

The text is online. You will need to make an account and purchase access from the text provider, Zybooks.

1. Sign in or create an account at learn.zybooks.com
2. Enter zyBook code: OKSTATECS1113CrickFall2021
3. Subscribe

Grading

- Participation activities: 10%
- Challenge activities: 10%
- Lab assignments: 30%
- Final independent project: 20%
- Exams: 30%
- You will have the opportunity to complete additional lab assignments for extra credit.

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.

Policies

- Readings include participation and challenge activities, which should be carried out as you read the online text. Each unit also includes a number of lab programming exercises. All of these are due on specific dates, ordinarily on the Monday after the period scheduled for covering the material.
- Throughout most of the course, work will be submitted through the online interface. However, you will have to install your own Java compiler and produce code independently for the final project.
- The project is a text-based two-player board or card game. After the midterm, you will receive instructions about the project expectations. The project will involve progressive submissions of increasingly-functional code over the second half of the course.
- There will be two exams, a midterm worth 10% and a comprehensive final accounting for 20% of your grade. You will be responsible for installing the Respondus lock-down browser (available on the course’s Canvas page) and providing a webcam and microphone for online proctoring. The Respondus browser is only available for Windows and Macintosh systems; if you are using another OS such as Linux, you may take a live proctored online exam with Prof. Crick, provided at a specific time to be arranged.
• Academic integrity is taken very seriously. You are permitted (and indeed encouraged) to discuss the course material with fellow students in general terms on the Canvas discussion board, 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**, 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.

### Class schedule

- **Weeks 1-2 (Aug 16 - Aug 27):** Introduction to Java
- **Weeks 3-4 (Aug 30 - Sep 10):** Variables / Assignments
- **Weeks 5-6 (Sep 13 - Sep 24):** Branches
- **Weeks 7-8 (Sep 27 - Oct 8):** Loops
- **Week 9 (Oct 11 - Oct 15):** Midterm
- **Week 10-11 (Oct 18 - Oct 29):** Arrays
- **Week 12-13 (Nov 1 - Nov 12):** User-defined methods
- **Week 14-16 (Nov 15 - Dec 3):** Objects and Classes, Fall Break
- **Week 17 (Dec 6 - Dec 10):** Final exam

### Due dates

- **Aug 30:** Unit 1 coursework due
- **Sep 13:** Unit 2 coursework due
- **Sep 27:** Unit 3 coursework due
- **Oct 11:** Unit 4 coursework due
- **Oct 15:** Midterm due
- **Oct 22:** Final project milestone (board printout) due
- **Nov 1:** Unit 5 coursework due
- **Nov 5:** Final project milestone (player input) due
- **Nov 15:** Unit 6 coursework due
- **Nov 19:** Final project milestone (computer moves) due
- **Nov 29:** Unit 7 coursework due
- **Dec 3:** Final project due
- **Dec 10:** Final exam due