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: Mondays 12:30-2:00 and 4:00-6:30, online at https://meet.jit.si/CrickCSOfficeHours. Drop by to join the 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. Live office hours are held every week, and you are encouraged to attend with questions.
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: OKSTATECS1113CrickSpring2023
3. Subscribe

Grading

- Participation activities: 10%
- Challenge activities: 10%
- Lab assignments: 35%
- Final independent project: 25%
- Exams: 20%

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 Tuesday 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 game such as Checkers. A few weeks into the class, 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 one exam, 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). 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 (Jan 17 - Jan 27): Introduction to Java
• Weeks 3-4 (Jan 30 - Feb 10): Variables / Assignments
• Week 5-6 (Feb 13 - Feb 24): Branches
• Week 7-8 (Feb 27 - Mar 10): Loops
• Week 9 (Mar 13 - Mar 17): Fall break
• Week 10-11 (Mar 20 - Mar 31): Arrays
• Week 12-13 (Apr 3 - Apr 14): User-defined methods
• Week 14-15 (Apr 17 - Apr 28): Objects and Classes, Fall Break
• Week 16 (May 1 - May 5): Finish final project
• Week 17 (May 8 - May 12): Final exam

### Due dates

• Jan 31: Unit 1 coursework due
• Feb 14: Unit 2 coursework due
• Feb 28: Unit 3 coursework due
• Mar 21: Unit 4 coursework due
• Mar 31: Final project milestone (board printout) due
• Apr 4: Unit 5 coursework due
• Apr 14: Final project milestone (player input) due
• Apr 18: Unit 6 coursework due
• Apr 28: Final project milestone (computer moves) due
• May 2: Unit 7 coursework due
• May 5: Final project due
• May 12: Final exam due