

# INTRODUCTION TO SOFTWARE SYSTEMS

Spring 2024 (Section 1727)

---

<b>Instructor:</b>	Joshua Crotts	<b>Time:</b>	MW 4:45PM – 6:00PM
<b>Email:</b>	ljcrotts@iu.edu	<b>Place:</b>	Luddy Hall (IF) 1106

---

**Office Hours:** TBD in Luddy 3025Q

**Textbook/Course Notes:** Joshua Crotts, *Teaching Java: A Test-Driven Approach*. PDF in Canvas.

## Ancillary Resources:

1. Cay Horstmann, *Big Java: Late Objects*, 2e.
2. Y. Daniel Liang, *Introduction to Java Programming and Data Structures*, 13e.
3. Mark Allen Weiss, *Data Structures and Problem Solving Using Java*, 4e.
4. Joshua Bloch, *Effective Java*, 3e.

**Student Learning Outcomes:** Upon completion of this course, students should be able to:

1. Use basic programming constructs including variables, decisions, loops, and functions in writing programs;
2. Utilize object-oriented programming concepts of classes, objects, encapsulation, inheritance, and polymorphism in designing programs;
3. Demonstrate skill in using software development methodology to design, implement and debug object-oriented software systems;
4. Use appropriate components from the Java Collections framework to solve a given problem;
5. Use an IDE to develop, debug, test and run programs;
6. Analyze searching and sorting algorithms using asymptotic analysis techniques.

**Prerequisites:** CSCI-C 211 or CSCI-H 211 or CSCI-C 200 or CSCI-H 200 or equivalent

**Catalog Description:** Design of computer software systems and introduction to programming in the environment of a contemporary operating system. Topics include a modern object-oriented programming language; building and maintaining large projects; and understanding the operating system interface.

## Grading Policy:

Assignments .....	40%
Labs .....	10%
Midterm .....	25%
Final .....	25%

## Important Dates:

Midterm .....	Wednesday, February 28, 2024
Final Exam .....	TBD

**Attendance Policy:** Lecture attendance, at least in my section, is not required. *If you are sick, do not come to class!* Lab attendance, on the contrary, *is mandatory*. If you do not show up to lab, you will receive a zero on that lab's assignment. You can attend another section's lab if and only if you email all lab instructors involved. Submitting the lab from home is not acceptable. In the event that you are sick or have some school event to attend, you can complete it from home/outside of class, but it must be finished by the Friday of the assigned week **and** you must email all lab instructors. Submitting it, then emailing after the fact, results in a zero.

**Assignment Corrections Policy:** If you score below a 90% on an assignment, you can earn *up to 90%* credit by doing assignment corrections. Corrections means redoing the questions that you got wrong initially, whether that means rewriting an algorithm, fixing comments/style, and so forth. You will submit these to a separate autograder link after your initial assignment is due.

**Lateness Policy:** Each student has three “late day” assignment submissions that are no-questions-asked; i.e., you can use them as needed throughout the semester. These are already set up in the autograder. From here, assignments are automatically marked down 25% up to 48 hours after the assignment is due. After this, no submissions are allowed, nor can you do corrections on an assignment that receives a zero due to lateness. Assignment corrections must be submitted on-time no matter what; any corrections submitted after their deadline will not be applied.

**Academic Integrity & Collaboration Policies** Don't cheat. You're all college students; you know what this means. All students must write their own code. To be a bit more clear, though:

Here is what you *can* do in this class:

- Discuss any and all lecture material and content with other students.
- Discuss labs and assignments with other students. This includes looking directly (i.e., not through a digital medium) at other students' code. Paired programming is allowed and encouraged, so long as you swap the “Driver” and “Navigator” roles from time-to-time.
- Look up high-level concepts on websites such as StackOverflow. E.g., “how do I write an if statement that short-circuits?”
- Ask for help, with code, from the instructors. You can also post code to the class Discord.

Here's what you *cannot* do in this class:

- Use ChatGPT or any other assistant bot that writes code for you. Don't do it. It's confidently incorrect most of the times (trust me, we know from experience!).
- Post code to external websites where people solve the question for you. This includes Chegg, StackOverflow, and so forth.
- Look up the exact question we ask on a lab, assignment, or otherwise. E.g., “Write the `celsiusToFahrenheit` method, which receives a temperature in Fahrenheit and converts it to Celsius.”
- Copying code, verbatim, from another student.
- Doing partial work. For instance, if Joe does problems 1-4 on an assignment and his friend Carol does problems 5-8, they cannot merge their problems together to create a submission.

Any incidents of academic misconduct will result in a zero on the assignment or worse, depending on the severity. Cheating on an exam will result in a zero in the course. If you have any questions on what does or does not constitute an academic integrity violation (or you wish to report an incident), please let your course instructor know.

**Elasticity Statement:** It is the intention of the instructor that this syllabus and course calendar will be followed as outlined; however, as the need arises there may be adjustments to the syllabus and calendar. In such cases, the instructor will notify students in class and via e-mail with an updated syllabus and calendar within a reasonable timeframe to allow students to adjust as needed. Ignorance of the policies written in this syllabus is not an excuse for not following them.

**Tentative Course Outline:**

<b>Week</b>	<b>Monday</b>	<b>Wednesday</b>	<b>Lab</b>	<b>Assignment</b>
1 (1/8)	Intro, Variables, & Datatypes	Static Methods & Testing	IntelliJ & Setup	A1 Out (1/8)
2 (1/15)	MLK Jr. Day ( <b>NO CLASS</b> )	Strings & Conditionals	Basic Programs	
3 (1/22)	Strings & Conditionals cont.	Methods (Helpers & Recursion)	Methods	A1 Due (1/22) A2 Out (1/22)
4 (1/29)	Loops	Loops cont.	Loops	
5 (2/5)	Arrays	ArrayLists	Arrays, ArrayLists	A2 Due (2/5) A3 Out (2/5)
6 (2/12)	Java Collections API	More Collections & Generics	Collections & Generics	A1 <sub>C</sub> Due (2/12)
7 (2/19)	Streams	Streams cont.	Streams & Review	A3 Due (2/19) A4 Out (2/19)
8 (2/26)	Review	<b>Midterm</b>	<b>NO LAB</b>	A2 <sub>C</sub> Due (2/26)
9 (3/4)	Objects & Classes	Objects & Classes cont.	Basic OOP	A3 <sub>C</sub> Due (3/4)
10 (3/11)	Spring Break ( <b>NO CLASS</b> )	Spring Break ( <b>NO CLASS</b> )	<b>NO LAB</b>	
11 (3/18)	Mutation & Aliasing	Mutation & Aliasing cont.	More OOP	A4 Due (03/18) A5 Out (03/18)
12 (3/25)	Interfaces	Inheritance	Even More OOP	
13 (4/1)	Inheritance & Abstract Classes	The “ASPL” Interpreter	Even Even More OOP	A5 Due (4/1) A6 Out (4/1)
14 (4/8)	Unchecked Exceptions	Checked Exceptions & I/O	Exceptions & I/O	A4 <sub>C</sub> Due (4/8)
15 (4/15)	Searching & Algorithm Analysis	Sorting & Algorithm Analysis	Sorting & Searching	
16 (4/22)	Concurrency	Final Exam Review	<b>NO LAB</b>	A6 Due (4/22) A5 <sub>C</sub> Due (4/22)