CMPINF 0401: Intermediate Programming 01

Spring, 2025

COURSE DESCRIPTION

This course is designed to deepen students' understanding of programming concepts using the Java language. Building on foundational programming skills, students will explore intermediate topics such as object-oriented design, data structures, algorithms, exception handling, and file I/O. Through hands-on coding exercises and projects, students will develop the skills necessary to design and implement more complex software applications. The course emphasizes problem-solving, code efficiency, and software design principles, preparing students for more advanced programming challenges in Java.

INSTRUCTOR

Hao Qin, Room 527
Email: hao.qin@scupi.cn
Office Hours: Tuesday 3:30-5:30 pm, Thursday 3:30-5:30 pm, or by appointment

LECTURES

Tuesday & Thursday 13:50-15:30
Location: S506 SCUPI Building. In-person attendance is required for all class meetings.

RECITATION

TBA

TEXTBOOK

Starting Out with Java, From Control Structures through Data Structures, latest edition, by Gaddis and Muganda [Publisher: Pearson]

COURSE OBJECTS AND LEARNING OUTCOMES

In this course, you will be expected to:

- Gain proficiency in applying intermediate-level programming concepts using Java.
- Understand and implement core object-oriented principles, such as inheritance, polymorphism, and encapsulation.
- Learn how to design and implement complex data structures (e.g., arrays, linked lists, trees, and hash maps).
- Develop algorithms for common problems and understand algorithmic complexity.
- Gain experience with exception handling, file input/output, and other advanced features in Java.
- Enhance problem-solving and critical thinking skills by working on real-world programming challenges.
- Collaborate on group programming projects to foster teamwork and communication skills in software development.

COURSE REQUIREMENTS AND GRADING

There will be homework assignments, quizzes, and two exams.

- Homework: 40%
- Quizzes: 10%
- Midterm Exam: 20%
- Final Exam: 30%

ABET OUTCOMES

- An ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics.
- An ability to function effectively on a team whose members together provide leadership, create a collaborative environment, establish goals, plan tasks, and meet objectives.
- An ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions.
- An ability to acquire and apply new knowledge as needed, using appropriate learning strategies.

HOMEWORK

Homework assignments and their due dates will be given in the lectures. All work should be clearly presented, showing all steps to demonstrate your understanding of the process. While collaboration with classmates is encouraged, the final submission must be your own independent work, and plagiarism is strictly prohibited. **No late homework** will be accepted.

QUIZZES

There will be short quizzes given during the class meetings. Quizzes are designed to test basic skills and prepare you for the exams.

EXAMS

There is a midterm exam and a final exam. Each test will focus on the material presented since the previous exam, while also having the potential to include content from earlier sections. Attendance at all exams is mandatory. Make-up exams will only be given in the event of an emergency and only if advance notification is provided.

ACADEMIC INTEGRITY

Maintaining academic integrity is essential in this course. All work submitted must be your own, whether individual or group assignments. Plagiarism, cheating, or any form of dishonesty will not be tolerated and will result in disciplinary action as per the institution's policies. You are encouraged to collaborate with classmates on understanding concepts, but all submitted work must reflect your independent effort. Properly cite any sources or assistance received and uphold the highest standards of academic honesty in all your work.

TENTATIVE PROGRESS

	Topic Sections
1	Java Basics
2	Input and the Scanner Class
3	Programming Errors
4	Switch Statement
5	Value Parameters
6	More References
7	Building Classes
8	Data Structures