

CMSC 132: Object-Oriented Programming II

Course Introduction



Department of Computer Science
University of Maryland, College Park

Course Catalog Description

- Introduction to use of computers to solve problems using software engineering principles
- Design, build, test, and debug medium-size software systems. Learn to use relevant tools
- Use object-oriented methods to create effective and efficient problem solutions
- Use and implement application programming interfaces (APIs)
- Programming done in Java

Things You Will Learn

- **Object-oriented software development**
 - Modern software development techniques
 - Object-oriented design
- **Algorithms & data structures**
 - Lists, trees, graphs
- **Programming skills**
 - Java API, IDE, testing, debugging

Course Is Not Just About Java

- May seem to focus on Java
 - All programming in Java
 - Many interesting Java language features
- Lessons intended to be general
 - Principles should apply to all languages
 - Ways of thinking about design
 - General ideas about software
 - Can translate skills to other languages

Course Is Not Just About Programming

- Software development involves a lot more than programming and debugging
- Developing software that doesn't satisfy your customer, or find a customer, is pointless
- Poor (or no) design will make it hard to modify or reuse your software
 - And you will have to modify it
- Lack of testing, plans, and build process leaves you lost, with no idea how to get back on track
- We have to cover a lot of programming ground, but we will also touch on these issues

Assume You Already Know

- **Coding**

- Variables, operators, loops, arrays

- **Basic object-oriented programming**

- Classes, methods, inheritance

- **Java**

- Class libraries, exceptions

- **Tools**

- Eclipse IDE, debugger

Where does 132 fit in?

- **CMSC 131**

- Basic programming skills

- **CMSC 132**

- Software design & basic algorithms

- **CMSC 212/213/216**

- Low-level programming

- **CMSC 250**

- Discrete math & logic

- **CMSC 351**

- Analysis of algorithms

Organization

- **Class Web Page**

<http://www.cs.umd.edu/class/summer2010/cmsc132/schedule.shtml>

- **Personnel**

 - **Coordinator**

 - **Nelson Padua-Perez**

- **Classes**

 - **Lectures**

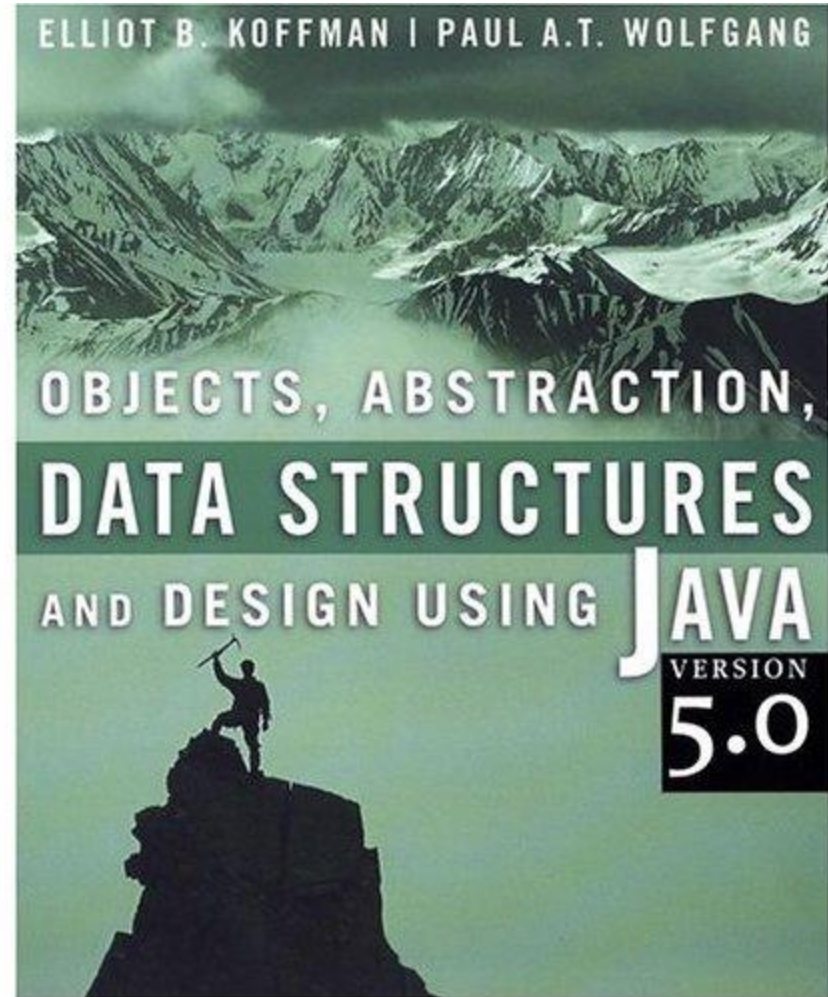
 - **Labs**

 - **Office hours**

Textbook

■ Recommended

- “Objects, Abstractions, Data Structures and Design Using Java (version 5.0)”
- By Elliot Koffman and Paul Wolfgang



Textbook (cont.)

- **Recommended**
 - **“Java Precisely (2nd Edition)”**
 - **By Peter Sestoft**



Projects

■ 5 projects

- Evaluate design, coding, testing skills
- Tries to involve interesting application areas
 - Networking, user interfaces, data compression

■ Late policy

- Projects due at 6 pm
- 20% penalty, up to 9am the next morning
- Plan to complete all projects on time

Projects (cont.)

- Environment

- Eclipse IDE

- Do not use your cmsc131 workspace (Create a new one).

- Your cmsc131 linuxlab account is no longer valid.

- Automated submission & testing

- Submit server

- <https://submit.cs.umd.edu>

- Maintains record of submissions

- CVS repository

- May use for research

- Release testing

- Can evaluate project using real test cases

Grading

- Based on
 - Projects, homework exercises, quizzes, midterms, final
- Point distribution (roughly)
 - 30% Projects (5)
 - 10% Homeworks
 - 15% Quizzes
 - 20% Midterm
 - 25% Final Exam
- Available on-line
 - <https://grades.cs.umd.edu>

Course Bulletin Board

■ Bulletin Board (Forum)

■ <https://forum.cs.umd.edu/forumdisplay.php?f=104>

■ Policy on project postings

■ Can ask about specification, setup, tools, etc.

■ Do **not** ask about design, implementation, etc.

■ Violators may face penalty for academic dishonesty

Facebook group

- Search for CMSC 132
- No project questions, etc.
- Just for fun/social

Academic Honesty

- All individual assignments & exams must be done individually (except "open" assignments)
- Do not copy (or allow others to copy) your work in any way
- Submissions will be compared to submissions from current and previous semesters
- Cases of academic dishonesty will be referred to the University's Office of Judicial Programs
- Visit Student Honor Council website for more detailed explanation of academic dishonesty

Excused Absences

- **Students must apply in writing and furnish documentary support for excused absences**
- **Support should explicitly indicate the dates or times the student was incapacitated**
- **Excused absence does not typically translate into project extensions**
- **Students requesting reasonable academic accommodations due to a disability must provide a letter from the Office of Disability Support Services**

Course Advice

- Start projects **early**
 - **make use of release testing if offered**
- Ask questions
- Read book
- Attend lectures
- Attend labs
- Attend office hours
- Pay attention to regrade deadlines

Topics Preview

■ Algorithms & data structures

- Asymptotic efficiency
- Lists, stacks, queues
- Trees, heaps
- Sets, maps, graphs
- Recursion

Topics Preview

- **Object-oriented software development**
 - **Software life cycle**
 - **Requirements & specifications**
 - **Designing objects & classes**
 - **Testing & code coverage**
 - **Programming paradigms**
 - **Design patterns**

Topics Preview

■ Programming skills

- Java collection framework

- Exceptions

- Threads, synchronization

- Java APIs

- Graphics User Interfaces (GUI)