Modern Programming Practices

Programming is the most basic part of Computer Science, as it is the basic language for expressing structures, processes, algorithms, systems, everything computable and computing related.

There are many different approaches and languages for programming, but currently object-oriented is the dominant model, and Java is the dominant language.

This course will provide an introduction to the OO paradigm, including analysis and design; to the use of UML to develop and communicate requirements and designs; and to implementation techniques, including an introduction to advanced features of the Java programming language.

Topics include:

- Objects and classes
- Analysis of requirements and development of design
- UML class, sequence, and object diagrams
- Relationship between UML diagrams and Java implementation
- Inheritance, interfaces and polymorphism
- Open-closed principle
- Collection processing with lambdas and streams
- Advanced unit-testing and exception-handling
- Annotations
- Programming with multiple threads
- Java generics and generic programming

Student Learning Outcomes. By the conclusion of MPP, the student will be able to demonstrate knowledge and skill in the following areas:

1. Ability to create use case, class, sequence, and object diagrams based on a detailed problem statement
2. Ability to translate UML diagrams into a full Java program
3. Ability to work with lambda expressions, method expressions, and their representation as nested classes
4. Ability to implement SQL-like queries on Collections in the form of Stream pipelines, making use of Java's rich Stream libraries
5. Beginning-level knowledge of Java Swing, Java generics, and generic programming
6. Understand the important role of the inward stroke of programming in the field of software development

Outcomes 1-5 will be the technical focus of the course; they will be addressed in lectures, in-class exercises, labs, exams, and in the course project. Outcome 6 will mainly be addressed experientially, by group meditation practice, but also through discussion of main points for each lesson.

Class Schedule

We will meet in the classroom in the morning sessions 10-12:30 Mon-Fri. The afternoon sessions will take place 1:30-3:30 pm. On Saturday mornings we will have only a morning session: 10-12:30.

Group meditations will take place at the end of the morning session (15 minutes) and at the end of the afternoon session (20 minutes).

Lecture Format. In the classroom, sessions will be conducted in a lecture format. In the afternoons, students will work on lab assignments unless there is some scheduled activity (such as going over lab solutions or taking a quiz); in that case, the format will be the same as for the morning session. In either case, students are expected to attend both morning and afternoon sessions Monday – Friday (attendance will be taken).

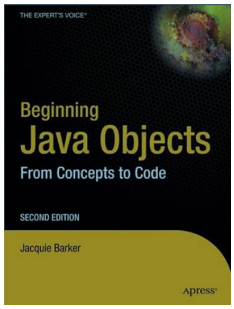
Course materials, including assignments, will be available through Sakai.

Exams will be conducted as per the schedule; laptops will not be required (or allowed) in the exam rooms, but you are encouraged to bring your laptop for the daily classroom sessions when you attend in-person.

On the first day of class, we will go through the structure of the course using Sakai.

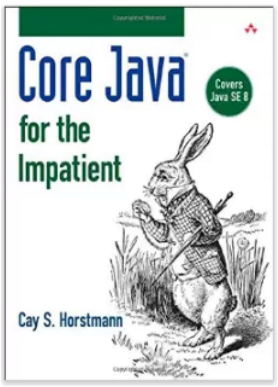
Required Course Text

- Jacquie Barker, *Beginning Java Objects: From Concepts to Code*, 2nd edition, Apress, 2005. Intro to OOAD.

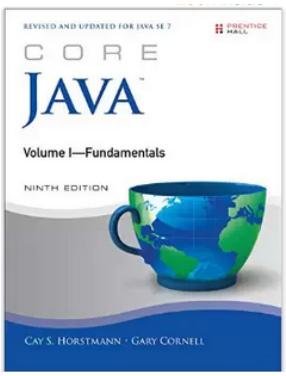


Additional Books and Resources (not required)

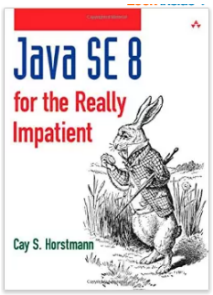
- Cay Horstmann, ***Core Java for the Impatient***, Addison-Wesley, 2015 (recommended) Covers Java 8



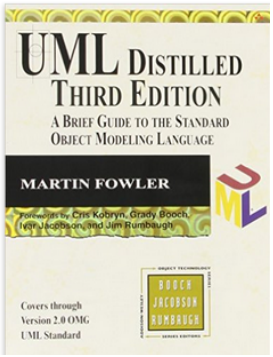
- Cay Horstmann**,** *Core Java Volume I--Fundamentals (9th Edition)*, Prentice-Hall, 2012. Covers Java 7, treatment is more thorough.



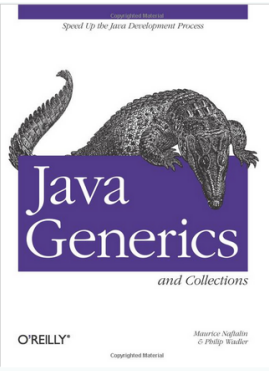
- Cay Horstmann, *Java SE 8 for the Really Impatient*, Addison-Wesley Professional, 2014. Covers JavaFX quickly.



- Martin Fowler, *UML Distilled, 3rd Edition,* UML syntax and best practices.



- Naftalin and Wadler, *Java Generics.* Introduction to Java generics.



- J. Bloch, *Effective Java,2nd Edition.* Best practices in Java design and implementation.

In-Class Exercises

During each lecture, a few exercises related to the lecture material will be mentioned. You will have 7-10 minutes to work on a given exercise, either individually or with one or two other students. These exercises will help you master the concepts of the course more quickly. All these exercises are stored in an Eclipse project named InClassExercises, which can (and should) be downloaded from Sakai onto your laptop. Some exercises are in the form of code; others are PDFs or JPGs; but all can be found in this project. When the time comes each day for an in-class exercise, be ready to read instructions and work with startup code located in this project. You will not submit your work on these, though some of the exercises are continued in the lab for that lesson. Details about setting up this Eclipse project are explained in the file Labsetup.pdf in the setup folder in Resources.

Labs and Presentation

Each lesson has a corresponding lab or "assignment". Some labs require Java code; others require creation of UML diagrams and other forms of documentation.

In addition to regular assignments, there will be a project during the second week of the course. The project will be done in groups. Groups consist of 3 persons. Group submissions are done through Sakai (see next section for details).

Each group will submit their project on the Monday of the third week (instructions for

submission will be announced). There will be an opportunity for a few groups to do project presentations for extra credit on the Saturday of the second week.

Submitting Your Labs

Instructions for each lab will be shown in a Sakai assignment ("Assignments" is an option in the left panel of your Sakai app). Each assignment will provide instructions and, if necessary, other materials. When you are ready to turn it in, please follow these instructions:

1. An assignment is submitted by attaching items (code, jpeg's, other documents) to the assignment page (you will see instructions for how to do this in the assignment itself).
2. *Submitting the project.* One person in your group will submit the work for your entire group. (Everyone in your group will get the same grade for this submitted work. There is no need for more than one person to submit a project.)

Professional Etiquette

You are expected to behave as you would in the context of a real IT job in the US. With Compro students, there are rarely any issues about professional etiquette, but you should be aware that you can lose up to 3 points (out of 100 total for the course) for failure to observe acceptable codes of behavior. Be alert to the following:

1. Attend class in the expected way
2. Don't be late to class
3. Don't leave class before it is time to leave
4. Don't skip class
5. Respectful attitude toward the professor. (Arguing about grades is a dangerous path to walk.)

Evaluation Criteria

The course grade will be determined according to the following (assuming that you pass the standard programming test given at the end of the block):

Activity	Percent Value
Labs	10%
Project	10% (12% e.b.)
Midterm Exam	40%
Final Exam	40%
Professional Etiquette	(worst: -3%)

Academic Honesty

Students are expected to submit only their own work (except for labs or other activities designated as group activities). During exams, they must not look at other students' work, discuss exam contents with other students at any time (including bathroom breaks), or attempt to access outside resources (such as internet or email). The academic dishonesty policy stated on the Compro website is reproduced here:

> **Academic Dishonesty:** Graduate students caught cheating will receive a grade of NC. A second case of cheating results in suspension from the university. Cheating includes copying from someone else as well as letting someone else copy your materials, or not following the policies during the test (e.g., not using a cell phone at any time; not having notes, etc).

Grading Scale

The following grading scale will be used in this course:

Range	Letter Grade
93 -100	A
90 - 92	A-
87 - 89	B+
83 - 86	B
80 - 82	B-
77 - 79	C+
73 - 76	C
70 - 72	C-
Below 70	NC