# CS 2110: Object-Oriented Programming and Data Structures

# Resources

Here are quick links to online resources that you may find valuable in this course.

## Website documents

- Student handbook
- Setup guide
- Style guide
- Transition to Java
- Input/output in Java

## Supplemental material

- JavaHyperText: online textbook originally prepared for CS 2110 by Cornell professor David Gries
- Object-Oriented Design and Data Structures: online course notes originally prepared for CS 2112 by Cornell professors Andrew Myers and Dexter Kozen
- The Java Tutorials: official tutorials from the creators of the Java language
- Principled Programming: free textbook by Cornell emeritus professor Tim Teitelbaum covering many of the algorithms in CS 2110

## References

- **Java 21 API documentation**: Specifications for types and methods in the Java standard library (bookmark this now!)
- The Java Language Specification: Formal specification of Java syntax and semantics (handy if you need to settle an argument or understand corner-case behavior, but the Java Tutorials are generally much more useful)

## Notes on other online resources

**Google search results**
Unfortunately, the top results are usually polluted by sites that specialize in “search engine optimization”. Their pages look relevant and engaging, but their content is mostly “fluff”—low on details and certainly not authoritative (not a citable source in a paper). Prefer hits from “oracle.com”—as the maintainers of Java, their API documentation and language specifications are authoritative.

**StackOverflow**
As a successful implementation of crowdsourcing, the top responses for common questions tend to be accurate, and the debates can be enlightening if you have sufficient background to follow them. But the quality of responses can vary significantly across questions, and the focus is on quick answers, not tutorials. As a result, SO tends to work better as an “expert resource” than an effective learning tool. SO responses are almost never authoritative themselves (exception: personal anecdotes by the people originally involved in the language’s development), but the better ones might cite an authoritative source.

**Wikipedia**
Content tends to be accurate and well-cited. The level of depth varies considerably between articles, but even thorough articles tend to work better as expert resources than tutorials. As with any encyclopedia, Wikipedia is not citable itself, but it tends to link to relevant references, some of which may be authoritative.

**ChatGPT**
Chatbot responses are never authoritative, never citable, often difficult to verify, and occasionally flat-out wrong. Additionally, using tools like ChatGPT and CoPilot to generate code for assignments is a violation of academic integrity in this course (remember: the purpose of these assignments is to *practice* programming; the resulting program isn’t really of value or interest on its own). It may have some use as a study aid, but you must verify your understanding with demo programs and authoritative references.

**Books**
Books from reputable publishers are fantastic! Their long-form, linear format is well-suited to learning, providing sufficient context to build your understanding on. Don’t worry if it was published a decade ago or more—the material in CS 2110 hasn’t changed much over those years. They may not be able to provide quick answers to specific questions like a search engine or chatbot can, but they will leave you with a broader and deeper understanding of their subject.

## How did Dr. Muhlberger learn this material?

- The Java Tutorials
- Java API documentation
- Books
- Practice!