Department of Computer and Information Sciences
COSC 455 – Programming Languages

Fall 2025

Instructor: Dr. Josh Dehlinger
Office: 7800 York Road, Room 476
Office Hours: T 3:00 - 4:00
By email and appointment

Email: jdehlinger@towson.edu
Slack: https://cosc455dehlinger.slack.com
Phone: 410.704.4536 (Slack/email preferred)

Course Hours: T 4:00 – 6:40
Course Location: YR 302
Course Website: Blackboard

Required Textbook: Programming Languages Pragmatics, 4th edition
Michael L. Scott, Morgan Kaufmann
ISBN # 9780124104099
Companion Website: https://booksite.elsevier.com/9780124104099/

References and other Suggested Readings:

- *Concepts of Programming Languages*, 9th Edition, Robert Sebesta
- *Why Prolog?* Gregory L. Lazarev
- *Programming Languages – Principles and Paradigms*, Allen Tucker and Robert Noonan
- *Essentials of Programming Languages*, 3rd Edition, Daniel Friedman and Mitchell Wand
- *Programming Language Pragmatics*, 2nd Edition, Michael L. Scott
- *Concepts, Techniques and Models of Computer Programming*, Peter Van Roy and Seif Haridi
- *Software Language Engineering*, Anneke Kleppe
- *Seven Languages in Seven Weeks*, Bruce A. Tate
- *Learn Rust https://www.rust-lang.org/learn*
- *The Rust Programming Language https://doc.rust-lang.org/book/*
- *Rust by Example https://doc.rust-lang.org/rust-by-example/*

Course Description: Underlying concepts in high-level programming languages and techniques for their implementation, a survey of a selected group of such languages along with a discussion of the interrelationship between programming and programming languages.

Prerequisite: COSC 336 – Data Structures and Algorithm Analysis

Course Objectives: After having finished this course, students will:

- Recognize the fundamental concepts that make up most high-level programming languages
- Understand the tradeoffs between language design and implementation
- Appreciate how/why concepts or features in different programming languages are selected
- Know the process involved in the design of a high-level programming language
- Be aware of data type definitions and implementations
- Understand the design of programming control structures
- Comprehend different programming language paradigms (e.g., functional, object-oriented, imperative)
- Realize the importance of functional programming in the context of parallel processing

Course Format: Active learning techniques, lectures, discussion sessions (in class AND online), presentations, lab exercises and projects may be used. ***Students are expected to read the textbook*** and find and use current content on the course subjects using the library, internet and provided resources. Some assignments will necessitate collaborative learning while others will require individual research and presentation.

Grading Policy: Students will be evaluated on the following basis:

Assignments / Labs:	20%
Projects:	40%
Exams:	40%

Final course grades will abide by the following scale:

A	95-100	**B-**	80-82.9	**D**	63-66.9
A-	90-94.9	**C+**	75-79.9	**D-**	60-62.9
B+	87-89.9	**C**	70-74.9	**F**	Below 60
B	83-86.9	**D+**	67-69.9		

Lab Assignments / Exercises: There will be several lab assignments/ exercises involving programming given throughout the semester. Lab exercises are meant to provide a "hands on" experience to students in order to emphasize the material recently covered during lecture and provide students opportunities to develop a toolset of software development skills/tools. Lab exercises are intended to be highly interactive in which students are encouraged to seek help to fully understand the material. All lab assignments / exercises will be given with ample time to complete. Students should start early and frequently check in their work to their GitHub account. Students are reminded that the instructor is available for help during the lab classroom and office hours, through Slack and email, and otherwise appointments. Lab exercises *cannot* be made up under any circumstances.

Projects: There will be 3-4 significant programming projects given throughout the semester. All projects will be given to provide students ample time to complete. Students should start early and frequently check in their work to the department Subversion server. Students are reminded that the instructor is available for help during office hours and otherwise through Slack, email or appointments.

Exams: There will be 2 exams for this course – a midterm and a final exam (each 20% weight). The date of the midterm exam will be announced during the semester. Make up exams can only be given in extenuating and documented circumstances (see below) and must be arranged in advance. The final exam will be given on *Tuesday, December 9, 2025, 5:15pm –7:15pm in YR 302*.

Late Policy: All assigned work (i.e., assignments and projects) is expected to be completed and submitted by the stated deadline. Late work may not be accepted or, at best, be penalized up to 10% off each day late. No questions will be answered, or assistance provided, on any graded item not submitted after the deadline has passed. Students are encouraged to discuss homework and projects as a means to share knowledge, experience and lessons learned as part of the learning process, but academic honesty should be strictly observed (see below).

Posting of Grades: University policy prohibits posting of grades in any form. The instructor will not report grades via email or in response to phone calls. Grades for the semester can be accessed online via Blackboard.

Cheating and Plagiarism: Academic honesty is strongly observed. This course may consist of both individual and team assignments. A team project is an assignment in which collaboration is allowed and highly encouraged. However, the work of the team *must* be of the team's creation and not plagiarized from other sources. Individual assignments *must* reflect the work of the individual student and of his/her creation. While studying together, discussion and collaboration is encouraged, individual assignments *must be individually prepared* – copying or

sharing files, diagrams and/or code is considered cheating. The penalty for cheating will, at a minimum, consist of a grade of zero for the dishonest work and may lead to the possibility of **course failure** depending on the severity. Students are responsible for reading and knowing Towson University's policy regarding academic dishonesty, located at https://www.towson.edu/about/administration/policies/03-01-00-student-academic-integrity-policy.html.

Artificial Intelligence Usage: Artificial intelligence (AI), large language models (LLMs) and other generative AI tools are awesome and will certainly be a tool of the future in your career(s). As such, in this course students are permitted (and, at times, encouraged) to use AI-based tools on some graded activities (e.g., assignments, quizzes, exams, projects, etc.). The instructions for each graded activity will include information about whether and how you may use AI-based tools to complete the assignment. All sources, including AI tools, must be properly cited. Use of AI in ways that are inconsistent with the parameters above will be considered violation of academic integrity.

Please note that AI/LLM results can be biased, inaccurate, or wholly hallucinated and should be viewed as a tool to help you complete certain tasks rather than a substitution for your understanding and mastery of the course concepts/topics. That is, you are still responsible for understanding the course material and, when warranted, providing hand-written solutions without the use of AI to demonstrate/assess mastery of the course concepts/topics. It is also your responsibility to ensure that the information you use from AI/LLMs are accurate and represent the material/technical concepts covered in class. Additionally, pay attention to the privacy of your data. Many AI/LLM tools will incorporate and use any content you share, so be careful not to unintentionally share copyrighted materials, original work, and/or personal information.

Learning how to thoughtfully and strategically use AI-based tools may help you develop your skills, refine your work, and prepare you for your future career. If you have any questions about citation or about what constitutes academic integrity in this course, please feel free to contact me to discuss your concerns.

Blackboard Website: There will be a Blackboard website for this course. Students will be responsible for frequently checking the site for updates and announcements. All course related materials will be available for download from the Blackboard site.

Computer Lab Information: Lab hours and policies for the COSC and Cook Library can be found at http://www.towson.edu/fcsm/departments/computerinfosci/resources/labs.html and here.

Classroom and Lab Policy: Food and drink is not allowed in the labs and classrooms with the exception of water in the classroom only. All cell phones should be turned off or put on silent to avoid disruptions and distractions.

Email Policy: All email correspondence with the instructor must be conducted using the student's Towson University email account or via Slack. The instructor will not read/respond to any email messages from outside accounts.

Repeat Policy: Students may not repeat a course more than once without prior permission of the Academic Standards Committee.

If you have a learning disability and/or need accommodation for any reasons, please advise the instructor as early as possible in the course.