COURSE: CMPS 344 – Programming Languages
Department of Computing Sciences, University of Scranton

DATE: Spring 2026 *(January 28, 2026 – May 22, 2026)*

INSTRUCTOR: P. M. Jackowitz

OFFICE HOURS: As posted (online and office door), and by appointment.
(Additionally, "Microsoft Teams" can be used for remote meetings.)

TELEPHONE: (570) 941-6107

EMAIL: paul.jackowitz@scranton.edu

WWW: https://www.scranton.edu/faculty/jackowitz/

Course Description: (Prerequisite: CMPS 240 *[CMPS 250 Recommended]*) A study of programming languages from both the theoretical and practical perspectives. A survey of major and developing paradigms and languages is undertaken which includes use of specific languages to broaden the student's experience. Implementation is studied through an introduction to language translation along with a study of run-time models and interfaces with virtual machines. (Undergraduate Catalog 2025-2026)

STUDENT LEARNING OUTCOMES: Upon completion of the course, a successful student will have the ability to do each of the following:

1. Knowledge and understanding of the evolutionary history of programming languages and of computing in general.
2. Knowledge of and experience with formal languages, including regular and context-free grammars; specifically as they pertain to the lexical and syntactic aspects of programming languages.
3. Knowledge of and experience with Attribute Grammars as a means of representing semantic elements of programming languages.
4. An in depth understanding of the process of language translation.
5. An appreciation of the concept of "binding" and its use in understanding many aspects of programming and programming languages.
6. Facility with the terminology of programming languages regarding matters such as abstraction, data types, expressions, statements, control flow and subprograms.
7. Exposure to selected relevant constructs present in a variety of programming languages, as a means of appreciating alternatives provided in different programming languages. Languages such as Ada, Eifel, Modula-2, etc. are typical.
8. Fluency in LISP programming as an example of the functional paradigm.
9. Exposure to the Prolog programming language as a means of appreciating the declarative and logic paradigms.

TEXTS: Programming Languages: Principles and Paradigms - Second Edition, Maurizio Gabbrielli and Simone Martini, Springer, 2023.

Course Materials: The corresponding Brightspace "course" will provide students access to course materials, with "drop boxes" configured for the submission of work for the instructor to evaluate.

GRADING:	Worth
Tests *(approximate date) and Quizzes (as announced)*	
Week of March 9th	20%
Final Exam: *(comprehensive, as per Final Exam Schedule)*	40%
Assignments: Submission and Assessment	40%

Attendance: *Your attendance at all class meetings is expected. Your participation during class, as evidenced in part by your attentiveness, speaking in class to ask and answer questions and participate in discussions, will be considered in the determination of the final course grade.*

PROCEDURES:

Lectures:

- please sit in the same seat for every class meeting
- feel free to ask and answer questions, and to contribute to discussions
- classroom use of all electronic devices/gadgets (including computers) is at the full discretion of the instructor. *(Distracting others or yourself will not be tolerated.)*

Tests and Quizzes:

- always announced in advance, and **no make-ups will be given**
- notice must be given if a test must be missed due to serious illness or emergency

Assignments:

- assignments are activities accomplished outside of class meetings that require the development and submission of specified items (typically source programs) that may be tested, evaluated and graded
- each student is required to submit the results they have accomplished for each assignment
- discussions and mutually beneficial collaboration among students is encouraged, but must not be to the point of representing someone else's effort and understanding as one's own, as this would be considered to be **academic dishonesty**
- although learning how to make effective use of online and AI resources has merit, overusing these to provide results instead of the student developing them themselves hinders understanding and would likewise be considered to be **academic dishonesty**
- academic dishonesty will be dealt with **severely;** see Academic Code of Honesty in the Student Handbook at: https://www.scranton.edu/academics/cte/acad-integ/acad-code-honesty.shtml
- each assignment will have a specified **due date** and results submitted **after the due date are considered to be "late"**, and may be assessed a late penalty.
- assignments typically have a corresponding specified **subsequent submission deadline**, after which results may not be submitted, and are considered to be **"too late"**
- incomplete results generally will receive a grade much higher than zero, while not submitting will result in a grade of zero

Other:

- See "Syllabi Language" regarding "My Reporting Obligations as a Required Reporter" at: https://www.scranton.edu/equity-diversity/faculty-resources.shtml