CS 280: Programming Language Concepts

Syllabus, Fall 2016

Professor Sean Cusack
scusack@njit.edu

Fall 2016 Office Hours: Tuesday 12-1, or by appointment

CS 280 - Programming Language Concepts
Conceptual study of programming language syntax, semantics and implementation. Course covers language definition structure, data types and structures, control structures and data flow, run-time consideration, and interpretative languages.

Please include CS280 and your section number in the Subject: line of any email you send; it will make it easier to manage my emails. I will do the same in emails I send to you.

Course Webpage: http://web.njit.edu/~gwryan/CS280

All course materials, including lecture notes, assignments and solutions, will be posted on the web page. Announcements and notices will be posted on http://web.njit.edu/~gwryan and sent by email.

Textbook:

Robert Sebesta, Concepts of Programming Languages, 11th Edition.
Pearson. ISBN 978-0-13-394302-3

Grading

Attendance	2%
Programs	40%
Midterm	25%
Final	33%

A student who does not submit two or more programming assignments will receive an F for this course.

Topics:

- Common features of programming languages
- Lexical Syntax
- Grammars
- Names

- Types
- Semantics
- Expressions
- Control Flow
- Subprograms
- Encapsulation
- Memory Management
- Event Handling
- Concurrency

Goals for the Course:

The student will be able to recognize similar features of different programming languages.

The student will have an easier time learning new programming languages.

The student will gain an appreciation of the strengths and weaknesses of different programming languages.

The student will demonstrate an ability to apply knowledge of computing and mathematics appropriate to the discipline.

The student will demonstrate an ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.

The student will demonstrate an ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.

The students will recognize the need for and an ability to engage in continuing professional development.

The student will demonstrate an ability to use current techniques, skills, and tools necessary for computing practice.

Ethical Conduct

Cheating during in-class tests or take-home examinations or homework is, of course, illegal and immoral. Programming assignments are NOT collaborative efforts. You may discuss problems with each other, but you may NOT copy lines of code from anybody or anywhere without attribution. You may NOT use code in your assignments that you did not write without attribution. Giving your work to someone else to copy from is just as much cheating as copying someone else's work.

The essential quality of the NJIT **University Code on Academic Integrity** is that each student shall demonstrate honesty and integrity in the completion of all assignments and in the participation of the learning process. Adherence to the University Code on Academic Integrity promotes the level of integrity required within the university and professional communities and assures students that their work is being judged fairly with the work of others. See http://www.njit.edu/academics/pdf/academic-integrity-code.pdf

Notes on Programming Projects:

- All programming assignments will be in C++. You will NOT need to submit a printout of your code.

- All programming projects must be submitted electronically via Moodle
  - Put SOURCE CODE ONLY into a SINGLE zip file
  - Upload the zip file into Moodle.
  - You can resubmit as often as you like.
  - Do not wait until the last minute.
  - Late submissions are penalized

- Every file should include a block of identifying comments at the very top of each file, as follows:
  - CS 280, the section number, and FALL 2016
  - Your name
  - Assignment #

- Programming projects will be graded on a scale of zero to 10

- Each day or part of a day that you are late reduces your grade by 1 point. Online submissions will be disabled five days past the due date.

- If your program does not compile, your grade will be a 1

- Partial answers or incorrect output will reduce your grade

- **A student who does not submit two or more programming assignments will receive an F for this course.**

Course Outline:

- Introduction [chapter 1 + 2]
- Lexical and Syntax Analysis [chapter 3 & 4]

- Names [chapter 5]
- Types [chapter 6]
- Expressions [chapter 7]
- Statements + Subprograms [chapter 8-10]

- Encapsulation and ADTs [chapter 11]
- Object Oriented Programming [Chapter 12]
- Memory Management
- Events
- Topics Survey
- FINAL EXAM

Note that exams are cumulative.

Important Dates

30-Sep	Program 1 Due
27-Oct	Program 2 Due
31-Oct	Common Midterm
17-Nov	Program 3 Due
14-Dec	Program 4 Due
16-Dec - 22-Dec	FINALS WEEK