CSC 252DL Course Syllabus

Dr. Chuck C. Liang

Office Hours: Mondays and Tuesdays 1-2pm, Thursdays 11am-12pm, or by email appointments. Office hours are both in-person in Adams 116 and Zoomed. Hours are subject to change: always check the Office Hours folder on blackboard for the lastet hours and zoom links.

Office Phone: (516 463) 5559

Email: chuck.c.liang@hofstra.edu (mailto:chuck.c.liang@hofstra.edu)

Information Specific to this Semester

There has been a distance-learning version of CSC252 since 2013. The undergraduate version of this class, csc123, will be taught in person and because of this, I am able to give you a zoom link to attend my live lectures synchronously, Tuesdays and Thursdays 4:20-5:45pm. Although lectures will be recorded, you are strongly encouraged to take advantage of this opportunity to improve your chances of passing the class. Experience during the pandemic shows that attending classes live on zoom is the next best thing to attending classes in person. However, you are asked to keep the zoom links confidential: **do not share them with anyone who's not in CSC252DL**.

Special Measures Against Cheating

Individuals caught cheating will face disciplinary actions from the university. **There will be no warning.** Individuals posting course materials and solutions online may also face legal action for violating the university and the instructor's copyrights. No course materials may be shared with others without consent from the instructor.

Interactive Assessment may be used. The instructor will have the right to require that you submit to an **oral exam** in a one-on-one setting before assigning you a final grade.

Course Description

A study of the semantics, specification and behavior of programming languages. The course will focus on various programming language paradigms including functional, imperative, object- oriented and aspect-oriented programming. Programming assignments using example languages from these paradigms will be required. Emphasis will be placed on learning languages such as Scheme, Perl, C#, ML, C++ and Rust. Other topics covered include language syntax, control structures, objects and functions.

Prerequisite: CSC 16, 17; CSC 14 and 24 strongly recommended

Recommended Text: "Programming Languages: Concepts and Constructs, 2nd Edition" by Ravi Sethi.

Tentative List of Topics

1. Introduction and review of programming concepts:

- Memory allocation
- Object oriented programming
- Type System
- Functional and class Abstraction
- Recursion

2. Mathematical Foundations: The Lambda Calculus
3. Formal Models of Computation:
4. Lazy versus Eager Evaluation: The Church-Rosser Theorem
5. Dynamic versus Lexical Scoping
6. Side-effects and State
7. Object Orientation
8. Types
    - the typed lambda calculus
    - Safe versus unsafe type systems
    - runtime versus static typing
    - various forms of polymorphism
    - principal types
    - inheritance
    - templates and generics
9. Typed functional programming
10. Zero Overhead Abstraction
11. Approaches to Memory safety

- garbage collection
- smart pointers in C++
- Rust

12. Aspect Oriented Programming.
13. Advanced topics (if time allow)

Exams, Assignments and Grading

Assignments will be given regularly. There will be a midterm exam and a final. The final exam will be cumulative. Periodic quizzes, including one-on-one quizzes may also be given. The grade distribution will be roughly 70% exams and quizzes and 30% attendance, programming assignments and other homeworks. Grading will be curved but the exact curve can be modified by the instructor. Students are required to keep copies of all programming assignments throughout the semester. When working in a group, all group members must possess current versions of the assignment. Final Note: The contents of this syllabus may be modified depending on the progress of the course.