CoSc 450 **Programming Paradigms** Fall, 2021

Programming Paradigms
Syllabus

Instructor

Stan Warford
Email: `Stan.Warford@Pepperdine.edu`
Office: RAC 112
Phone : 310-506-4332 (on campus, x4332)

Office hours

Monday, 11:00 – 11:50
Tuesday, 9:00 – 9:50
Thursday 1:00 – 1:50
Friday, 11:00 – 11:50
And by appointment

Course Web page

The course Web page will be used to post the assignments and late-breaking announcements. You are responsible for checking it regularly.

https://www.cslab.pepperdine.edu/warford/cosc450/

Objective

The objective of this course is to learn three major programming models that complement the procedural and object-oriented models: the functional, declarative/logic, and concurrent models. Each of these models is based on a body of theory and each can be implemented in various programming languages—in this course Scheme, Prolog, and Java respectively. Although this course introduces the theoretical basis of each model, the emphasis is on the practice of programming.

A paradigm is a pattern of thinking that is frequently difficult to change. For example, the procedural and object-oriented models are based on the concept of state, as exemplified by the ubiquitous assignment statement. It may be difficult to imagine programming without an assignment statement, but the pure functional and declarative/logic models have no assignment statement. In both of these models, you will have to make a paradigm shift to thinking without assignment statements as you contemplate various programming solutions. Similarly, the procedural model is concerned with a single thread of control. It will take a paradigm shift to think about several processors executing several threads of control concurrently to solve a single problem.

In this course, we will immerse ourselves completely in each of the three paradigms. That is, rather than discuss the ideas abstractly and concurrently we will attempt to experience them sequentially through programming. The course is thus divided into three distinct parts, one for each of the programming models in which we will program with each of the three programming languages. The goal is not to make you an expert programmer in any of the three languages. Each language has far too many advanced features and nuances to be learned in one third of a single semester. However, you should be equipped from this introduction to extend your expertise in each model.

Motivation

Here is a quote by John Prevost a Software Engineer for ArsDigita Corporation:

> A good background in different programming languages has given me the ability to think outside the box of one language style or another. Without this background, I would not be as valuable to my decidedly practically-minded company—even though we do not generally use theoretically interesting languages.

Learning outcomes

The computer science program learning outcomes (PLO) for the computer science/mathematics major are the ability to:

PLO 1. Implement algorithms
PLO 2. Prove computational theorems
PLO 3. Analyze computational systems
PLO 4. Communicate technical results

The course student learning objectives (SLO) for CoSc 450, Programming Paradigms is the ability to:

Implement an algorithm using selection and iterative control structures. (PLO 1)
Implement an algorithm using recursive control structures. (PLO 1)
Implement searching and sorting algorithms. (PLO 1)
Prove propositional and predicate logic theorems. (PLO 2)
Prove temporal logic theorems. (PLO 2)
Prove theorems using mathematical induction. (PLO 2)
Prove the correctness of a program from its formal specification. (PLO 2)
Implement an algorithm using a functional language. (PLO 1)
Implement an algorithm using a declarative language. (PLO 1)
Implement a concurrent algorithm. (PLO 1)

Required texts

There are four texts for this course, one of which is online. See the course webpage for details.

Final grade

16% Homework
52% Tests (26% each test)
32% Final - cumulative

Class schedule

The course web page has the schedule for the homework assignments, which are due twice weekly. The exam schedule is as follows:

Test 1, Monday, September 27
Test 2, Thursday, October 28
Final, Tuesday, December 14, 7:30 a.m. – 10:00 a.m.

Late Homework Policy

Written homework is due in my office by 5:00 pm on the due date. Programming homework is due on Courses/Sakai at 11:55 p.m. on the due date. Half credit for homework one assignment late. No credit thereafter. Partial submissions (that is, some problems on time and some others late for half credit) are not allowed. You will receive liberal partial credit, so it is better to turn in an incomplete attempt than to turn in for late credit.

Programming Homework Policy

If your program does not compile (build)
- Automatic 20% of total points for that assignment
- No resubmission

Late submission
- Accepted up to the time of the following assignment
- Notify me via email
- Automatic 50% of graded score

Late submission that does not compile
- Automatic 10% of total points for that assignment

Course evaluations

Completion of the online course evaluation is a requirement for this course. They are anonymous and can be viewed by the professor only after final grades have been submitted to the registrar. Email a proof of completion to me before the final exam.

Attendance policy

Attendance is important and may affect your final grade. You are responsible for making sure that your attendance has been recorded. Please provide written documentation for excused absences. There will be no makeup exams. If you miss an exam due to illness or an unexpected major emergency, the final exam score will be substituted for your missed exam score. Doctor's note required for all missed exams.

Accessibility notice

Any student with a documented disability (physical, learning, or psychological) needing academic accommodations should contact the Student Accessibility Office (Main Campus, Tyler Campus Center 264, x6500) as early in the semester as possible. All discussions will remain confidential. Please visit https://www.pepperdine.edu/student-accessibility/ for additional information.

Academic integrity

See https://seaver.pepperdine.edu/academics/academic-support/integrity/ for the academic integrity standards at Seaver College.

Mission support

See https://www.pepperdine.edu/about/our-story/mission-vision/ for the mission statement of the university and https://seaver.pepperdine.edu/about/our-story/seaver-mission/for the mission statement of Seaver College. This course supports these mission statements by investigating the truth of its discipline and by preparing students for lives of service to others in the field of computer science.