San José State University
Department of Computer Science
CS 152, Programming Language Paradigms, Section 2, Fall 2022

Instructor(s):	Tazmina Sharmin
Email:	tazmina.sharmin@sjsu.edu
Office Hours:	Tuesday 10:15 - 11:15 am
Class Days/Time:	Tuesday 9 - 10:15 am | Thursday 9 - 10: 15 am
Classroom:	Online via Zoom
Prerequisites:	CS 151 or CMPE 135 (with a grade of "C-" or better)

Course Description

Programming language syntax and semantics. Data types and type checking. Scope, bindings, and environments, compilers and interpreters, lambda calculus, recursion. Imperative vs. declarative languages. Functional and logic programming paradigms, and comparison to other paradigms. Hands-on introduction to Python, Prolog, and Scheme.

Course Format

The course will be conducted online over Zoom. All instruction will be delivered online, and all homework assignments and exams will be turned in electronically. All students must have access to a personal computer and have reliable Internet access in order to participate in this course.

Canvas Course Site

Course materials such as syllabus, lecture notes, assignments and exams can be found on the Canvas Leaning Management System course website at http://sjsu.instructure.com/. You are responsible for regularly checking with Canvas to learn of any updates.

Course Goals

Upon successful completion of this course, students will be able to understand programming language design. Students will gain working hands-on knowledge of the following programming languages: Python, Prolog, JavaScript, and Scheme.

Course Learning Outcomes (CLO)

Upon successful completion of this course, students will be able to:

1. Understand the concepts and terms used to describe languages that support the imperative, functional, object-oriented, and logic programming paradigms.
2. Acquire a high-level view of programming language concepts
3. Solve problems using the functional paradigm.
4. Solve problems using the object-oriented paradigm.
5. Solve problems using the logic programming paradigm.
6. Critically evaluate what paradigm and language are best suited for a new problem.

Textbook

This class does not require a mandatory textbook. Google is your friend!

Optional textbook (I will not be teaching by them but may use some reference from these books):

1. (TYSiFD) Dorai Sitaram. Teach Yourself Scheme in Fixnum Days. 2015. Available online for free at https://ds26gte.github.io/tyscheme/.
2. • (TAoP) Sterling and Shapiro. The Art of Prolog, Second Edition: Advanced Programming Techniques, 1993. Available online at The Art of Prolog; click on "Open Access" and then click "View on MIT Press Direct" in order to download a free copy of the book.

   Additional Readings TBD.

We will be using Python, Prolog, and Scheme in this class. Appropriate environments will need to be installed. I will outline a few options in my slides and notes for how to run code in those languages. However, if a student already has a favorite way to do that they don't need to follow my suggestions.

Course Requirements and Assignments

"Success in this course is based on the expectation that students will spend, for each unit of credit, a minimum of 45 hours over the length of the course (normally three hours per unit per week) for instruction, preparation/studying, or course related activities, including but not limited to internships, labs, and clinical practica. Other course structures will have equivalent workload expectations as described in the syllabus."

There will be 6-7 homework assignments. Homework assignments will be posted and submitted on Canvas. For full credit, they must be submitted by the posted due date and time. All assignments will be individual work. You may not share or copy code or answers from fellow students or from the web. If someone else copies your work, with or without your permission, you will be held responsible.

There will be one team project. You may choose to work by yourself or with a partner. For a team project, the work must be done by both team members and both team members will receive the same grade.

There is a final and two midterms. The final exam is worth 20% of the total grade for the class.

Grading Information

Homework: 30%
Project: 20%
Midterm: 30%
Final: 20%

Late Work

Late assignments will be accepted with a 1-point penalty for each day late. Late days include weekend days. For example, an assignment due on Monday by 5 PM will incur a penalty of 1 point if submitted at 8AM on Tuesday. No submissions will be accepted more than 3 days late.

Grade Scale

The letter grade will be determined based on the following scale:
A+ = 98% - 100%
A = 92% - 97%
A- = 90% - 91%
B+ = 87% - 89%
B = 83% - 86%
B- = 80% - 82%
C+ = 77% - 79%
C = 73% - 76%
C- = 70% - 72%
D = 60% - 69%
F = below 59

Classroom Protocol

- During live Zoom lectures, please keep your microphones muted during instruction, except when I am taking questions or while I am hosting a discussion.
- Please be respectful in all communication in this course. This includes the lectures and all online communication.
- Please keep up with all communication in this course, whether it is through email or through Canvas.
- Limit your distractions/avoid multitasking. You can make it easier to focus on the meeting by turning off notifications, closing or minimizing running apps, and putting your smartphone away (unless you are using it to access Zoom).
- Attendance is not mandatory and it does not form any part of your grade. But students are highly encouraged to participate at each class lecture. There will be no make-ups for missed midterm or assignments, unless any special arrangements are made with the instructor beforehand or there is an emergency. The student is responsible for any material he/she may have missed.
- Email to be sent to the instructor's SJSU email ID (tazmina.sharmin@sjsu.edu) only. I check email periodically during the day but much less during weekends. Please do not expect quick turnaround time during weekends.
- Start on your homework early and stay on top of them. Some assignments take way more time than you expect.
- Attend all sessions. From past semesters, data shows that there is a positive correlation between attendance and your overall grade.
- Start on your homework early and stay on top of them. Some assignments take way more time than you expect. Don’t let your initial impression fool you.
- Be prepared to learn A LOT. Some of this may require you to self-study certain topics. I will guide you through this journey but the onus of getting the best of this class lies on you.
- Attend all sessions. From past semesters, data shows that there is a positive correlation between attendance and your overall grade.
- Class sessions will not be recorded and students are not allowed to make audio / video recording or photography in class session without prior permission of the instructor.

University Policies

Per University Policy S16-9, university-wide policy information relevant to all courses, such as academic integrity, accommodations, etc. will be available on Office of Graduate and Undergraduate Programs' Syllabus Information web page at http://www.sjsu.edu/gup/syllabusinfo/ . Make sure to review these policies and resources.

Important dates:

August 19 – First day of instructions

August 29 - Last Day to submit prerequisite proofs

September 15 – Last Day to Drop Classes without a "W" Grade and Last Day to Add Classes via MySJSU

December 6 - Last Day of Instruction

CS 152 Programming Paradigms, Fall 2022, Course Schedule

Please note that I reserve the right to change the course schedule. I will communicate all changes to the course schedule via Canvas and email.

Week	Date	Topics
1	8/23//2022	Course introduction & Logistics
1	8/25/2022	Procedural Programming and Structured Programming
2	8/30/2022	Parsing, Context-Free Grammars, and Abstract Syntax Trees
2	9/1/2022	Parsing with ANTLR
3	9/6/2022	Operational semantics and Lambda Calculus
3	9/8/2022	Functional Programming and An Introduction to the Scheme Programming Language
4	9/13/2022	Scheme: higher order functions, recursion and efficiency
4	9/15/2022	Closures and Scoping, Macros
5	9/20/2022	Building an Interpreter
5	9/22/2022	Functional Data Abstraction

6	9/27/2022	Memory Management
6	9/29/2022	Midterm-1 review
7	10/4/2022	Midterm-1
7	10/6/2022	Object-Oriented Programming
8	10/11/2022	Inheritance and Polymorphism
8	10/13/2022	Type Systems, Static and Dynamic Typing
9	10/18/2022	Introduction to Logic Programming
9	10/20/2022	Prolog
10	10/25/2022	Resolution and Unification in Prolog
10	10/27/2022	Lists in Prolog and debugging Prolog
11	11/1/2022	Introduction to Smalltalk
11	11/3/2022	Metaprogramming in Smalltalk
13	11/08/2022	Midterm-2 review
13	11/10/2022	Midterm-2
14	11/15/2022	Ruby
14	11/17/2022	Ruby Blocks
15	11/22/2022	Virtual Machines
15	11/24/2022	Thanksgiving
16	11/29/2022	Dynamic code evaluation, taint analysis, and information flow analysis
16	12/01/2022	Python and IDE plugins
17	12/06/2022	Final Review