San José State University
Department of Computer Science
CS152, Programming Language Paradigms, Sections 1 and 2, Spring, 2023

Course and Contact Information

Instructor:	Saptarshi Sengupta, PhD
Office Location:	MacQuarrie Hall 416
Telephone:	408-924-4808
Email:	saptarshi.sengupta@sjsu.edu
Office Hours:	Thursday, 10:30 AM – 12:30 PM
Class Days/Time:	MW 3:00 PM-4:15 PM (**Section 1**), MW 4:30 PM-5:45 PM (**Section 2**)
Classroom:	MacQuarrie Hall 225
Prerequisites:	Object Oriented Design and Programming CS 151 or CMPE 135 (with a grade of "C-" or better in each of the classes)

Course Format

Faculty Web Page and MYSJSU Messaging

Course materials such as syllabus, handouts, notes, assignment instructions, etc. can be found on Canvas Leaning Management System course login website at http://sjsu.instructure.com. You are responsible for regularly checking with the messaging system through MySJSU at http://my.sjsu.edu (or other communication system as indicated by the instructor) to learn of any updates.

Course Description

Discussions on Programming Languages and Programming Paradigms, Computer Architecture, Turing Completeness, Data Storage, Data Types and Type Checking, Scope, Bindings, Environments, Compilers and Interpreters, Lambda Calculus, Recursion. Imperative vs. Declarative languages. Practical introduction to Python, Prolog, ML, JavaScript, Haskell and Scheme.

Course Learning Outcomes (CLO)

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

- Have a good understanding of the history of programming languages.
- Have a good understanding of computer architecture, data types, scope, typing, syntax and semantics.
- Have a good understanding of the different types of programming Languages, their design and constructs.
- Have a good understanding of parsing systems.
- Understand differences between compiled and interpreted languages.
- Have a good understanding of imperative programming paradigms: procedural, object oriented, and parallel.
- Have a good understanding of declarative programming paradigms: logic, functional, dataflow, database.
- Have working knowledge of Python.
- Have working knowledge of Prolog.
- Have working knowledge of ML.
- Have working knowledge of JavaScript.
- Have working knowledge of Haskell.
- Have working knowledge of Scheme.
- Understand lambda calculus.
- Understand recursion.

Required Texts/Readings

Textbook

None required.

Other Readings (Optional)

Programming Languages: Principles and Practice, 3rd edition 2012
Authors: Kenneth Louden and Kenneth Lambert
Publisher: Cengage Learning
ISBN-13: 978-1-111-52941-3

Other technology requirements / equipment / material

We will be using Python, Prolog, ML, JavaScript, Haskell and Scheme in this class. Appropriate environments may need to be installed. You are free to use your own environment setup and IDEs in addition to instructor recommended ones.

Course Requirements and Assignments

SJSU classes are designed such that in order to be successful, it is expected that students will spend a minimum of forty-five hours for each unit of credit (normally three hours per unit per week), including preparing for class, participating in course activities, completing assignments, and so on. More details about student workload can be found in University Policy S12-3 at http://www.sjsu.edu/senate/docs/S12-3.pdf.

Homework, Quizzes, Exams and a Final Project are expected for this class. Homework is due on Canvas by midnight on the due date. Each assigned problem requires a solution and an explanation (or work) detailing how you arrived at your solution. Cite any outside sources used to solve a problem. When grading an assignment, I may ask for additional information.

NOTE that University policy F69-24 at http://www.sjsu.edu/senate/docs/F69-24.pdf states that “Students should attend all meetings of their classes, not only because they are responsible for material discussed therein, but because active participation is frequently essential to insure maximum benefit for all members of the class. Attendance per se shall not be used as a criterion for grading.”

Final Examination or Evaluation

The final project presentations may be administered in person/online.

Grading Information

- Programming Assignments: 20%
- Quizzes: 20%
- Exam 1: 20%
- Exam 2: 20%
- Final Project: 20%

Note that "All students have the right, within a reasonable time, to know their academic scores, to review their grade-dependent work, and to be provided with explanations for the determination of their course grades." See University Policy F13-1 at http://www.sjsu.edu/senate/docs/F13-1.pdf for more details.

Determination of Grades

Semester grade will be computed as a weighted average of the scores obtained in each of the five categories listed above. No make-up tests or quizzes will be given, and no late homework (or other work) will be accepted except in extraordinary circumstances. Also, in-class work must be completed in the section that you are enrolled in.

Nominal Grading Scale:

Percentage	Grade
97 – 100 plus	A+
93 – 96	A
90 – 92	A-
87 – 89	B+
83 – 86	B
80 – 82	B-
77– 79	C+
73 – 76	C
70 – 72	C-
67 – 69	D+
63 – 66	D
60 - 62	D-
0-59	F

Classroom Protocol

- **Cheating** will not be tolerated.
- Student must be respectful of the instructor and other students. For example, No disruptive or annoying talking.
- Turn off cell phones
- Class begins on time
- Valid picture ID required at all times

University Policies (Required)

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.

CS152 / Principles and Paradigms of Programming Languages, Spring 2023, Course Schedule
The schedule is subject to change with fair notice communicated via Canvas course page or in class

Course Schedule

Week	Date	Topics, Readings, Assignments, Deadlines
1	01/25	Introduction to CS 152
2	01/30	Foundations and Computer Architecture
2	02/01	Computer Architecture (contd..)
3	02/06	Syntax and Semantics
3	02/08	Syntax and Semantics
4	02/13	Functions, Function Implementation and Memory Management
4	02/15	Language Design Criteria, Data Types, Abstract Data Types and Modules
5	02/20	Expressions and Statements, Procedures and Environments
5	02/22	Programming Paradigms (Basics, Turing completeness, Compiled vs. Interpreted)
6	02/27	Programming Paradigms (Compilers and Compiled Languages, Interpreters and Interpreted languages)
6	03/01	Programming Paradigms (other language classifications, major paradigms, paradigm vs. language, execution model)
7	03/06	**Exam 1**
7	03/08	Imperative Programming Paradigm (Procedural, Object oriented, Parallel): Procedural
8	03/13	Imperative Programming Paradigm (Procedural, Object oriented, Parallel): Object oriented
8	03/15	Imperative Programming Paradigm (Procedural, Object oriented, Parallel): Object Oriented and Parallel
9	03/20	Introduction to Python
9	03/22	Declarative Programming Paradigm: Logic programming
10	03/27	**Spring Recess – no classes**
10	03/29	**Spring Recess – no classes**
11	04/03	Introduction to Prolog
11	04/05	Declarative Programming Paradigm: Functional programming
12	04/10	Declarative Programming Paradigm: Functional programming
12	04/12	Lambda Calculus and Functional Programming in Python
13	04/17	Functional Programming in ML
13	04/19	Functional Programming in Haskell
14	04/24	Introduction to JavaScript
14	04/26	Introduction to Scheme

Week	Date	Topics, Readings, Assignments, Deadlines
15	05/01	**Exam 2**
15	05/03	Dataflow and Database Programming, Concurrent Programming, Correctness
16	05/08	**Final Project Presentations**
16	05/10	**Final Project Presentations**
17	05/15	Wrap up