SJSU SAN JOSÉ STATE UNIVERSITY

College of Science · Computer Science

Programming Paradigms Section 02

CS 152

Spring 2025 In Person 3 Unit(s) 01/23/2025 to 05/12/2025 Modified 01/28/2025

Contact Information

Instructor: Saptarshi Sengupta, PhD

Office Location: MacQuarrie Hall 416

Telephone: 408-924-4808

Email: saptarshi.sengupta@sjsu.edu

Office Hours: Friday, 11:00 AM – 1:00 PM

Class Days/Time: TuTh 3:00PM - 4:15PM

Classroom: BBC107

Course Description and Requisites

Programming language syntax and semantics. Data types and type checking. Scope, bindings, and environments. Functional and logic programming paradigms, and comparison to other paradigms. Extensive coverage of a functional language.

Prerequisite: CS 151 or CMPE 135 (with a grade of "C-" or better); Allowed Majors: Computer Science or Software Engineering; or instructor consent.

Letter Graded

Classroom Protocols

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

Program Information

Diversity Statement - At SJSU, it is important to create a safe learning environment where we can explore, learn, and grow together. We strive to build a diverse, equitable, inclusive culture that values, encourages, and supports students from all backgrounds and experiences.

Course Goals

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

1. Understand programming language design.
2. Achieve competence in functional and logic programming languages.

Course Learning Outcomes (CLOs)

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

- Recognize the history of programming languages
- Understand the concepts and terms used to describe languages that support the imperative, functional, object-oriented, and logic programming paradigms
- Explain the roles of interpreters, compilers, and virtual machines
- Read and produce context-free grammars
- Write recursive-descent parsers for simple languages
- Understand computer architecture, data types, scope, typing, syntax and semantics
- Understand differences between compiled and interpreted languages
- Write interpreters for simple languages that involve arithmetic expressions, bindings of values to names, and function calls
- Solve problems using the imperative paradigm
- Solve problems using the object-oriented paradigm
- Solve problems using the functional paradigm
- Solve problems using the logic programming paradigm
- Critically evaluate what paradigm and language are best suited for a new problem

Course Materials

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 programming languages spanning multiple paradigms in this class. Appropriate environments may need to be installed. You are free to use your own development environment setup 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.

The exams will contain several questions spanning multiple-choice, true/false, short answer, coding and mathematical proof types. Exams are open book and open notes. The exams should be done individually. No make-up exams will be administered except in the case of verifiable emergency circumstances.

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."

Grading Information

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

University Policies

Per University Policy S16-9 (PDF) (http://www.sjsu.edu/senate/docs/S16-9.pdf), relevant university policy concerning all courses, such as student responsibilities, academic integrity, accommodations, dropping and adding, consent for recording of class, etc. and available student services (e.g. learning assistance, counseling, and other resources) are listed on the Syllabus Information (https://www.sjsu.edu/curriculum/courses/syllabus-info.php) web page. Make sure to visit this page to review and be aware of these university policies and resources.

Course Schedule

CS152 / Principles and Paradigms of Programming Languages, Spring 2024, 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	01/23	Introduction to CS 152
1	01/28	Foundations of Programming and Computer Architecture
2	01/30	Computer Architecture (contd..)
2	02/04	Computer Architecture (contd..)
3	02/06	Syntax and Semantics
3	02/11	Language Design Criteria, Data Types, Abstract Data Types and Modules
4	02/13	Expressions and Statements, Procedures and Environments

Week	Date	Topics, Readings, Assignments, Deadlines
4	02/18	Functions, Function Implementation and Memory Management
5	02/20	Programming Paradigms (Basics, Turing completeness, Compiled vs. Interpreted)
5	02/25	Programming Paradigms (Compilers and Compiled Languages, Interpreters and Interpreted languages)
6	02/27	Programming Paradigms (other language classifications, major paradigms, paradigm vs. language, execution model)
6	03/04	Imperative Programming Paradigm (Procedural, Object oriented, Parallel): Procedural
7	03/06	Imperative Programming Paradigm (Procedural, Object oriented, Parallel): Object oriented
7	03/11	Imperative Programming Paradigm (Procedural, Object oriented, Parallel): Object Oriented and Parallel
8	03/13	Exam 1
8	03/18	The Functional Paradigm: Lambda Calculus
9	03/20	The Functional Paradigm: Lambda Calculus (contd..)
9	03/25	Functional programming using Scheme
10	03/27	Functional programming using Scheme (cont..)
10	04/01	Spring Recess - no classes
11	04/03	Spring Recess - no classes

Week	Date	Topics, Readings, Assignments, Deadlines
11	04/08	Functional programming using Haskell
12	04/10	Functional programming using Haskell (contd..)
12	04/15	Functional programming using Python (continued ..)
13	04/17	Functional programming using Python (continued ..)
13	04/22	Functional programming using JavaScript
14	04/24	Functional programming using JavaScript (continued ..)
14	04/25	Declarative Programming Paradigm: Logic programming with Prolog
15	04/27	Logic programming with Prolog (contd..)
15	04/29	Dataflow and Database Programming, Concurrent Programming, Correctness
16	05/01	Exam 2
16	05/06	Final Project Presentations
17	05/08	Final Project Presentations