

CSC 299 Concepts of Programming Languages

3 cr.

Instructor: TBA

email: TBA@salemstate.edu

Office: location

Office Hours: days and times

Phone: (978) 542-extension

Section	Time	Room	Final Exam
nn	days and times	location	Date and time

Catalog description:

In this course students will study principles underlying how programming language features are defined, composed, and implemented. Additionally, models underlying different programming languages, effective use of languages, and an appreciation of their limitations will be explored. Fundamental programming language paradigms such as, imperative, functional, object-oriented, and logic programming are presented. Furthermore, programming language translation, static program analysis, type systems, and memory allocation and management strategies are studied and discussed. Three lecture hours per week, plus programming work outside of class.

Prerequisite: CSC 260

Goals:

- CG01:** to gain a good level of understanding of fundamental programming language concepts;
- CG02:** to learn the effective use of programming languages, and appreciation of their limitations;
- CG03:** to gain basic knowledge of programming language translation and static program analysis;
- CG04:** to provide additional experience in problem-solving and programming in a number of different programming languages of different paradigms;
- CG05:** to enhance skills in problem analysis and program design and implementation;

Objectives:

Upon completion of this course, students will have demonstrated the ability to:

- CO01:** distinguish syntax and parsing from semantics and evaluation;
- CO02:** describe, design and apply different data types, including primitive and compound data types;
- CO03:** describe examples of different expressions and assignment statements;
- CO04:** discuss benefit and limitations of dynamic memory management and apply associated techniques dynamic memory management;
- CO05:** understand and use Abstract Data Types and encapsulation constructs in different programming languages;
- CO06:** design and implement solutions to use different programming paradigms to solve given problems;

Student Outcome (SO) vs. Course Objectives matrix

Student Outcome	CO01	CO02	CO03	CO04	CO05	CO06
SO-1	✓	✓	✓	✓	✓	✓
SO-2		✓		✓	✓	✓
SO-3		✓	✓		✓	✓
SO-4						
SO-5						
SO-6						✓

Notes:

- SO-1:** Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
- SO-2:** Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
- SO-3:** Communicate effectively in a variety of professional contexts.
- SO-4:** Recognize professional responsibilities and make informed judgments in computing practice based on legal and ethical principles.
- SO-5:** Function effectively as a member or leader of a team engaged in activities appropriate to the program's discipline. Apply computer science theory and software development fundamentals to produce computing-based solutions.
- SO-6:** Apply computer science theory and software development fundamentals to produce computing-based solutions.

Topics:**5****Preliminaries****PL5(0, 1, 0), PL9(0, 0, 0.5)**

The concepts of programming Languages.
 Major Programming Domains
 Major Influences on Language Design
 Various Categories of Programming Languages

Describing Syntax and Semantics**PL7(0, 0, 1), PL8(0, 0, 1), PL15(0, 0, 0.5)**

Lexical Analysis
 General Parsing Problem
 Complexity of Parsing
 Implantations Techniques

Data Types**PL4(1, 3, 0), PL11(0, 0, 1), PL14(0, 0, 1)**

The concept of a data type
 Characteristics of common primitive data types
 Designs of enumerations and subrange types
 Structured Data types (arrays, associative arrays, records, lists, and unions)
 The categories of data types
 Investigation of type checking, strong typing, and type equivalence rules
 Implantation methods of data types

Expressions and Assignment Statements**PL5(0, 1, 0)**

Arithmetic Expressions
 Semantic Rules and the order of the Evaluation
 Overloaded Operators
 Relational and Boolean Expressions
 Short-Circuit Evaluation
 Mixed-Mode Assignment

Dynamic Memory Management**PL10(0, 0, 2)**

Heap and Its Implementation
 Allocation and Deallocation of Dynamic data objects
 Garbage collection
 Continuous reference-count garbage collection concurrent
 Garbage collection

Abstract Data Types and Encapsulation Constructs**PL1(1, 1, 0), PL3(0, 1, 0)**

The Concept of Abstraction
 Design Issues for Abstract Data Types
 Parameterized Abstract Data Types
 Encapsulation Constructs

Concurrent Programming Paradigm**PL13(0, 0, 2)**

Types of Concurrency
 Program Dependency and Automatic Parallelization
 Task and Data Parallelism
 Communicating sequential processes
 Memory models for concurrency
 Concurrent programming constructs

Functional Programming Paradigm

PL2(3, 4, 0)

Fundamental Ideas of Mathematical Functions
 Implementation Models for Functional Languages
 Integration with other Programming Paradigms

Logic Programming Paradigm

PL5(0, 1, 0), PL17(0, 0, 1)

Logic Programming Fundamentals
 Abstract Implementation Model
 Programming Using Prolog
 Extending logic Programming Paradigm
 Integration with other Paradigms

Object-Oriented Programming Paradigm

PL1(1, 1, 0)

Classes and Objects
 Class Hierarchy and Inheritance
 Visibility and Information Exchange Polymorphism and Type Conversion

Web and Multimedia Programming Paradigms

PL10(0, 0, 1)

Code and Data Mobility
 Web-based Programming
 Virtual Machines and Run-time interface

Other Programming Paradigms

PL3(0, 1, 0), PL16(0, 0, 1)

Event-based Programming
 Agent-based Programming
 High productivity Massive Parallel Programming Synchronous Languages

Scripting Languages

PL5(0, 1, 0)

Components of Scripting Languages
 Abstractions in Scripting Languages

Programming Assignments: There will be several programming assignments in which students will be required to implement appropriate design components. All programs must conform to departmental guidelines for design and implementation, and laboratory reports must conform to the written guidelines supplied by the instructor.

Exams and quizzes: There will be one mid-term exam and a comprehensive written two-hour final examination.

The course grade will be determined using the following approximate weights: programming assignments – 40%, examinations (midterm and final) – 40%, written homework - 20%.

Course Objective / Assessment Mechanism matrix

	Written Homework	Programming Assignments	Examinations
CO01	✓	✓	✓
CO02	✓	✓	✓
CO03	✓	✓	✓
CO04	✓	✓	✓
CO05	✓	✓	✓

CO06		✓	
------	--	---	--

Bibliography:

- Robert W. Sebesta.** Concepts of Programming Languages, eleventh edition Boston, Massachusetts: Addison-Wesley, **2016**.
ISBN-13: 978-0-13-394302-3
- Michael Scott.** Programming Languages, fourth edition. Morgan Kaufmann, San Francisco, California, **2016**.
ISBN-13: 978-0124104099
- Arvind Kumar Bansal.** Introduction to Programming Languages. Boca Raton, Florida: CRC Press, **2014**.
ISBN-13:9781466565159
- Mitchell, John Clifford.** Concepts of Programming Languages. Cambridge, England: Cambridge University Press, **2003**.
ISBN: 0-521-78098-5
- Ryan Standifer.** The study of programming languages. Prentice Hall, **1995**.
ISBN: 0-13-726936-6
- Gabbrielli, Maurizio, Martini, Simone.** Programming Languages: Principles and Paradigms

Academic Integrity Statement:

“Salem State University assumes that all students come to the University with serious educational intent and expects them to be mature, responsible individuals who will exhibit high standards of honesty and personal conduct in their academic life. All forms of academic dishonesty are considered to be serious offences against the University community. The University will apply sanctions when student conduct interferes with the University primary responsibility of ensuring its educational objectives.” Consult the University catalog for further details on Academic Integrity Regulations and, in particular, the University definition of academic dishonesty.

The Academic Integrity Policy and Regulations can be found in the University Catalog and on the University website (http://catalog.salemstate.edu/content.php?catoid=13&navoid=1295#Academic_Integrity). The formal regulations are extensive and detailed - familiarize yourself with them if you have not previously done so. A concise summary of and direct quote from the regulations: "Materials (written or otherwise) submitted to fulfill academic requirements must represent a student's own efforts". *Submission of other's work as one's own without proper attribution is in direct violation of the University's Policy* and will be dealt with according to the University's formal Procedures. *Copying without attribution is considered cheating in an academic environment - simply put, **do not do it!***

University-Declared Critical Emergency Statement:

In the event of a university-declared emergency, Salem State University reserves the right to alter this course plan. Students should refer to www.salemstate.edu for further information and updates. The course attendance policy stays in effect until there is a university-declared critical emergency.

In the event of an emergency, please refer to the alternative educational plans for this course, which will be distributed via standing class communication protocols. Students should review the plans and act accordingly. Any required material that may be necessary will have been previously distributed to students electronically or will be made available as needed via email and/or Internet access.

Equal Access Statement:

"Salem State University is committed to providing equal access to the educational experience for all students in compliance with Section 504 of The Rehabilitation Act and The Americans with Disabilities Act and to providing all reasonable academic accommodations, aids and adjustments. **Any student who has a documented disability requiring an accommodation, aid or adjustment should speak with the instructor immediately.** Students with Disabilities who have not previously done so should provide documentation to and schedule an appointment with the Office for Students with Disabilities and obtain appropriate services."

Note: This syllabus represents the intended structure of the course for the semester. If changes are necessary, students will be notified in writing and via email.