Ira A. Fulton Schools of Engineering
Arizona State University

Course Syllabus
Principles of Programming Languages

Course Description

Programming languages are one of the most important and direct tools for the construction of a computer system: in a modern computer, different languages are routinely used for different levels of abstraction. Programming language is important because it defines the relationship, semantics and grammar which allows the programmers to effectively communicate with the machines that they program.

Learning Outcomes

By the end of this short course, students at a broad level will be able to:

Objective 1: Understand the salient features in the landscape of programming languages
Objective 2: Understand the essence of defining concepts of programming languages, so to allow critical choice about the level of abstraction

Course Structure

After reviewing the course materials for these objectives, you will have the opportunity to take ungraded practice quizzes. This is to make sure you have mastered the learning objectives.

Once you have mastered all of the course learning objectives, you will be prepared to take the Final Exam. After passing the final exam with an 70% or better, you would have completed CSE 340: Operating Systems prerequisite for ASU's Master of Computer Science degree.

Course Materials

You do not need to purchase materials for this course.

Time Commitment Per Module

- Module 1 (4 hours)
- Module 2 (5 hours)
- Module 3 (4 hours)
- Module 4 (6 hours)
- Module 5 (8 hours)
- 120 minutes for Final Exam

Time estimates are based on content provided in the course; additional readings outside the course have not been factored in these time estimates.

Course Grade Breakdown

Course Work	Quantity	Percentage of Grade
Final exam	1	100%

Grade Scale

You will need to pass the final exam with an 70% or better, and successfully pass proctoring requirements for the final exam.

Module 1: Regular Expressions

- 1.0: Overview
- 1.1: Lexical Analysis: Language and Syntax
- 1.2: Lexical Analysis: Regular Expressions
- 1.3: Lexical Analysis: Concatenation
- 1.4: Lexical Analysis: Kleene Star
- 1.5: Lexical Analysis: Regular Expression Examples
- 1.6: Lexical Analysis: Longest Matching Prefix Rule
- 1.7: Practice Quiz

Module 2: Syntax and Analysis Parsing

- 2.0: Overview
- 2.1: Introduction to Syntax Analysis
- 2.2: Context Free Grammars
- 2.3: Leftmost & Rightmost Derivations
- 2.4: Parse Tree
- 2.5: Ambiguous Grammar
- 2.6: Efficient Parsing
- 2.7: First Sets
    - 2.7.1: Example – First Sets
- 2.8: Follow Sets
    - 2.8.1: Example – Follow Sets
- 2.9: Conditions for Predictive Recursive Descent Parser
- 2.10: Writing a Predictive Descent Parser
- 2.11: Practice Quiz

Module 3: Basic Semantics and Pointer Semantics

- 3.0: Overview
- 3.1: Pointer Semantics Examples
- 3.2: Memory Allocation and Deallocation
- 3.3: Memory Errors
- 3.4: Practice Quiz

Module 4: Lambda Calculus

- 4.0: Overview
- 4.1: Introduction to Lambda Calculus
- 4.2: Disambiguation Rules
- 4.3: Semantics
- 4.4: Currying
- 4.5: Equivalence and Renaming
- 4.6: Substitution
- 4.7: Execution
- 4.8: Practice Quiz

Module 5: Type Systems

- 5.0: Overview
- 5.1: Introduction to Type Systems
- 5.2: Name Equivalence
- 5.3: Structural Equivalence
- 5.4: Structural Equivalence Algorithm
- 5.5: Hindley-Minler Type Interface
- 5.6: Implicit Polymorphism Part - 1
- 5.7: Implicit Polymorphism Part - 2
- 5.8: Practice Quiz

Course Completion

- Final Exam Proctoring Setup
- Practice Proctored Exam
- Final Exam