

COURSE DESCRIPTION

Department and Course Number	CS 405	Course Coordinator	Bryant
Course Title	Programming Languages	Total Credits	3

Current Catalog Description

Formal syntax and semantics; compilers and interpreters; virtual machines; representation of data types; sequence and data control; type checking; run-time storage management; functional, logic, and object-oriented programming paradigms; concurrency and exception handling.

Textbooks

Concepts of Programming Languages, 7th ed., by Robert W. Sebesta, Addison-Wesley Longman, 2006.

References

Modern Compiler Implementation in Java: Basic Techniques, 2nd ed., by Andrew W. Appel and Jens Palsberg, Cambridge University Press, 2002.

Course Goals

The goal of this class is for students to become knowledgeable about the theory and practice of programming language design and implementation.

Prerequisites by Topic

Algorithms and Data Structures and Automata and Formal Language Theory (with grade of "C" or better in both)

Major Topics Covered in the Course

Concepts of Programming Languages and their Implementation; Formal Syntax and Compiling; Denotational Semantics and Functional Programming; Axiomatic Semantics and Logic Programming; Names, Bindings, Type Checking, and Scopes; Data Types; Expressions; Statements; Subprograms; Abstract Data Types; Object-Oriented Programming; Concurrency; Exception Handling

Laboratory projects (specify number of weeks on each)

Functional Programming in Lisp (1 week)

Logic Programming in Prolog (1 week)

Design and Implementation of an Interpreter for a Programming Language Using the Formal Specification of the Language as a Reference (14 weeks)

Estimate CSAB Category Content

	CORE	ADVANCED		CORE	ADVANCED
Data Structures	<u>2</u>	_____	Computer Organization and Architecture	_____	_____
Algorithms			Concepts of Programming Languages	<u>36</u>	_____
Software Design	<u>2</u>	_____			

Oral and Written Communications

None

Social and Ethical Issues

None

Theoretical Content

The primary theoretical content is in the formal syntax and semantics of programming languages. Approximately 10 hours are spent on these topics.

Problem Analysis

All course sections require a project to implement a programming language interpreter from the formal specification of that language. The formal specification must be thoroughly analyzed before the interpreter can be designed and implemented.

Solution Design

All course sections require the design of an interpreter for a programming language. This design is normally divided into different aspects of language implementation, such as lexical analysis, syntax analysis, symbol table construction, type checking, intermediate code generation, and interpretation.