

## COURSE DESCRIPTION

Department and Course Number	<b>CS 401 (formerly CS 405)</b>	Course Coordinator	<b>Barrett Bryant</b>
Course Title	<b>Programming Languages</b>	Total Credits	<b>3</b>

### 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 multi-threading.**

### Textbooks

***Concepts of Programming Languages*, by Robert W. Sebesta, 8th ed., Addison-Wesley Longman, 2008.**

### References

***Modern Compiler Implementation in Java: Basic Techniques*, by Andrew W. Appel and Jens Palsberg, 2nd ed., 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  
Automata and Formal Language Theory**

### 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
	0	1		0	0
Data Structures	0	2	Computer Organization and Architecture	10	22
Algorithms			Concepts of Programming Languages		
Software Design					

## Oral and Written Communications

Every student is required to submit at least 0 written reports (not including exams, tests, quizzes, or commented programs) of typically 0 pages and to make 0 oral presentations of typically 0 minutes duration. This includes only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

## Social and Ethical Issues

Topics that address the social and ethical implications of computing are not covered in any course sections.

## 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

Please describe the design experiences common to all course sections.

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.