

# Course Information Sheet

## CSCI 4500

### Programming Languages

#### **Brief Course Description** (50-words or less)

In this course we will explore modern programming languages and the paradigm -- procedural or imperative, functional, and logic programming -- that each strives to accommodate. Projects involve at least three languages to get a feel for the language paradigms

#### **Extended Course Description / Comments**

We will cover both past and present languages, with an emphasis on modern programming languages. We will look at a wide spectrum of styles of programming that include imperative, scripting, functional, logic and object oriented languages and evaluate their strengths and limitations.

Specific topics include:

- Syntax and semantics.
- Names, binding and scope.
- Imperative, functional, logical based and object oriented paradigm.
- Types.
- Control flow.
- Programming: Functional, Scripting and Logical Programming.

#### **Pre-Requisites and/or Co-Requisites**

CSCI 1302: Software Development (Pre-Requisite)  
CSCI 2720: Data Structures (Co-Requisite)

#### **Required, Elective or Selected Elective**

Selected Elective Course

**Approved Textbooks**  
(if more than one listed, the textbook used is up to the instructor's discretion)

Author: Michael L. Scott  
Title: *Programming Languages Pragmatics*  
Edition: 3 or later.  
ISBN-13: 978-0123745149 or later.

Author: Robert W. Sebesta  
Title: *Concept of Programming Languages*  
Edition: 9 or later  
ISBN-13: 978-0131395312 or later

#### **Specific Learning Outcomes (Performance Indicators)**

At the completion of this course students should be able to do the following:

1. Explain the differences between imperative, functional and logical paradigms.
2. Explain why it is important to understand these programming language paradigms.
3. Explain when (and why) one paradigm is more applicable than another paradigm.
4. Create a lexer (using a tool like flex or lex) for a simple language.
5. Create a simple parser (using a tool like bison) for simple language.
6. Create and design a program using a functional programming language.
7. Create and design a program using a logical programming language.
8. Create and design a program using a scripting language

9. Demonstrate comprehension of short programs written in functional, imperative and logic paradigms.
10. Explain and evaluate design and implementation features of programming languages.

**Relationship Between Student Outcomes and Learning Outcomes**

		Student Outcomes										
		a	b	c	d	e	f	g	h	i	j	k
<b>Learning Outcomes</b>	1		●							●	●	
	2									●	●	
	3		●							●	●	
	4		●	●						●		●
	5		●	●						●		●
	6		●	●						●		●
	7		●	●						●		●
	8		●	●						●		●
	9		●							●	●	
	10		●	●						●	●	

**Major Topics Covered**  
(Approximate Course Hours)

3 credit hours = 37.5 contact hours  
 4 credit hours = 50 contact hours

Note: Exams count as a major topic covered

- Overview of Programming Languages (4 hours)
- Programming Language Paradigms (4 hours)
- Programming Languages Syntax and Semantics (4-hours)
- Scanning in Practice (4-hours)
- Parsing in Practice (4-hours)
- Functional Languages (lazy evaluation, evaluation order, higher order functions, currying, closures, static & dynamic scope, side- effects, introduction to LISP like languages, LIPS or Scheme and modern mainstream functional programming languages like Clojure, Groovy and Scala) (8 hours)
- Polymorphism (4 hours)
- Control Flow (4 hours)
- Names, Binding, Scope (4 hours)
- Scripting (4 hours)
- Data types (4 hours)
- Logical Languages (4 hours)

**Course Master**

Dr. Maria Hybinette