

## COURSE DESCRIPTION

<b>Course No.</b>	CSC 339	<b>Course Title</b>	Concepts of Programming Languages
<b>Course Type</b>	Required		
<b>Sem. Hours</b>	3	<b>Coordinator</b>	Lixin Fu

### Current Catalog Description:

Concepts of block-structured, object-oriented, functional, logic, and concurrent programming languages. Comparative study of syntactic and semantic features of these languages and writing programs using them.

### Textbook:

Concepts of Programming Languages, 10th Edition. Robert W. Sebesta, Addison-Wesley, 2009. ISBN: 978-0-13-139531-2.

### References:

C++ How To Program, by Deitel & Deitel, 5th edition, 2005.

### Course Outcomes:

Upon successful completion of this course, a student should be able to:

1. *describe* different paradigms through a comparative study of syntactic and semantic features
2. *develop* C++ programming skills by understanding pointers, memory management, and special C++ features such as templates and multiple inheritance
3. *obtain* a flavor of programming in other paradigms like functional and logic programming (Lisp and Prolog) which are quite different from "usual" imperative ones (like C++ or Java)

### Activities Enabling Program Outcomes (PO<sub>x</sub> refers to program student outcome x)

*Instruction:* This course focuses on the design and implementation of various programming languages. Computing theory, and in particular context free grammars and BNF notation, are discussed in application to language definition (PO<sub>a</sub> and PO<sub>j</sub>). Tradeoffs in expressability, readability, and ease of use are discussed (PO<sub>j</sub>). Discussions include how programming language selection affects the design and evaluation of computer-based systems (PO<sub>c</sub> and PO<sub>k</sub>). Languages include those that are common in current practice, including traditional languages and newer languages such as Python (PO<sub>i</sub>).

*Student Activities and Assessment:* Every offering of this course will include (details of assessment criteria and expectations are in outcome rubrics):

- One or more program/assignment showing translation of theory into practice (POa)
- One or more program/assignment or test question in which students select an appropriate language or programming paradigm to solve a problem (POj)

**Prerequisites by Topic:**

<p>Students must have</p> <ul style="list-style-type: none"> <li>• a grade of at least C in CSC 330 (Advanced Data Structures)</li> </ul>
---

**Major Topics Covered in the Course:**

<ul style="list-style-type: none"> <li>• Describing syntax and semantics</li> <li>• Imperative programming: constructs, data types, statements, procedures</li> <li>• Object-oriented programming: abstract data types, inheritance, object-oriented programming in C++, C#, and Java</li> <li>• Concurrent programming: threads in Java, principles of concurrency, concurrency in Ada</li> <li>• Functional programming</li> <li>• Logic programming</li> </ul>
---

**Estimated Curriculum Category Content (Semester hours):**

<i>Area</i>	<i>Core</i>	<i>Advanced</i>	<i>Area</i>	<i>Core</i>	<i>Advanced</i>
<b>Algorithms</b>	0	0	<b>Software design</b>	0	0
<b>Data structures</b>	0	0	<b>Prog. Languages</b>	3	0
<b>Comp Org &amp; Arch</b>	0	0			