

# CSC 461 Programming Languages

## SDSM&T Course Syllabus for Spring 2011

**Course:** CSC 461 Programming Languages (3 credits)  
**Prerequisite:** CSC 300 Data Structures  
**Room:** McLaury 313  
**Time:** MWF 11:00-11:50AM  
**Website:** <http://www.mcs.sdsmt.edu/csc461>

**Instructor:** Dr. Weiss  
**Office:** McLaury 311  
**Phone:** 394-6145  
**Email:** john.weiss@sdsmt.edu  
**Office Hours:** MTWTh noon-1:00PM, or by appointment

**Course Catalog Description:** CSC 461 Programming Languages (3-0) 3 credits. Prerequisite: CSC 300 with a “C” or better. This course consists of two parts. The first part introduces how programming languages are designed, including an introduction to the concepts of parsing and compiling. Issues related to implementation such as type checking, binding, and memory management are discussed. Secondly, the course will survey the spectrum of programming languages paradigms, including traditional imperative, object oriented, functional, and logic languages.

**Textbook:** *Concepts of Programming Languages, 9<sup>th</sup> ed.* (Sebesta, Addison-Wesley, 2009)

### Course Requirements

1. programming assignments (3)	30%
2. problem sets (3)	15%
3. midterm exam	22%
4. final exam (May 4, 8-10AM)	33%

### Topics

1. Introduction to concepts of programming languages	(1 week)
2. Formal languages, grammars, finite automata	(1 week)
3. Concepts of parsing and compiling	(1 week)
4. Data abstraction	(1 week)
5. Control structures	(1 week)
6. Expressions	(1 week)
7. Modules, binding, scope, parameter passing	(1 week)
8. Exception handling	(½ week)
9. Concurrency	(1 week)
10. Block structured languages	(½ week)
11. Scripting languages (Python)	(1½ weeks)
12. Object-oriented programming (Python, C++)	(2 weeks)
13. Functional programming (Lisp)	(2 weeks)
14. Software design and modeling languages (UML)	(½ week)

---

---

## Prerequisites

CSC 461 is an advanced computer science course. To succeed in this class, you should have experience writing substantial computer programs, with a good working knowledge of recursion, pointers, dynamic memory management, fundamental data structures and algorithms, and object-oriented programming (including class inheritance) in C++. Your mathematics background should include courses in college algebra and finite structures.

---

## Software

Practical exposure to different programming paradigms is an important part of this course. Be prepared to learn at least two new languages this semester. The fundamentals of syntax and semantics will be introduced in class, but teaching yourself to program in a new language is part of the CSC 461 experience.

Software and electronic reference materials for this course will be provided free of charge. Links to language processors and tutorials for selected languages (Python, Lisp, etc.) will be posted on the course website. In most cases, you will have the choice of working on either Windows or the Linux operating system on your home PC. Software for CSC 461 is preinstalled in the MCS Linux Lab in McLaury 215, and Linux accounts are available for all students enrolled in this course.

---

## Objectives

The primary objective of this course is to give the student an introduction to the theory and practice of programming languages. From a theoretical standpoint, we will discuss topics such as formal languages, various programming paradigms, data abstraction, control mechanisms, binding and scope, parameter passing, etc. From a practical standpoint, we will consider prototypical examples of different programming language paradigms, including block structured (C, Fortran, Pascal), concurrent (Java, Python), object oriented (C++, Java, Python), and functional programming (Lisp).

## Outcomes

Upon completion of this course, students will obtain the following outcomes:

- historical perspective on programming languages, from 1950's to the present
  - ability to critically evaluate the suitability of a programming language for a specific task
  - understanding of formal language theory and automata
  - understanding of formal methods of programming language syntax and semantics
  - understanding of fundamental programming language elements (data abstraction, control structures, expressions, exception handling)
  - understanding of issues related to programming language modules (binding, scope, parameter passing)
  - understanding of major programming language paradigms (procedural, object-oriented, functional, logic, concurrent)
  - experience writing programs with different programming language paradigms
  - greater understanding of the software development process
  - experience working in teams
-

---

## Grading

Letter grades will be assigned at the end of the semester, based on the weighted scoring system outlined above. Note that programming assignments and homework problems have a significant impact upon your final grade. In general, you will be allowed to work in teams of two on these assignments. To pass the course, you must successfully complete these assignments as well as pass the exams.

There is no “late policy” in this class. All assignments must be turned in by the due date, otherwise they will not be accepted. To receive full credit, programs must not only be correct, but must adhere to good programming style guidelines (standardized formatting, meaningful identifiers, modular code, good documentation, etc.). Program grading policies are further discussed in the Programming Guidelines document on the course Website.

---

## Academic Integrity

Although you may exchange ideas with your classmates, you must complete these assignments by yourself (or with members of your team, in the case of group projects). In particular, it is forbidden *under any circumstances whatsoever* to exchange source code with your classmates. **COPYING CODE IS A SERIOUS INFRINGEMENT UPON THE SDSM&T ACADEMIC INTEGRITY POLICY, AND WILL BE TREATED ACCORDINGLY.** Academic integrity is further discussed on the Academic Integrity policy statement on the course Website.

## Attendance

Attendance is required for all courses at SDSM&T. You are responsible for the lecture material as well as any assigned readings. Lecture material may diverge significantly from the assigned readings, so good attendance is particularly important in this class. Attendance is further discussed in the Classroom Conduct policy statement on the course Website.

## Electronic Devices Policy

Notebook computers may be used to take notes, but not for answering email, browsing the Web, or other non-course-related activities. No other electronic devices may be used during class time. Please be sure to turn off cell phones and pagers before class starts.

---

---

*The following statements must appear on all SDSM&T course syllabi:*

**ADA Statement**

Students with special needs or requiring special accommodations should contact the instructor (Dr. John Weiss, 394-6145) and/or the campus ADA coordinator (Jolie McCoy, 394-1924) at the earliest opportunity.

**Freedom in Learning Statement**

Under Board of Regents and University policy, student academic performance may be evaluated solely on an academic basis, not on opinions or conduct in matters unrelated to academic standards. Students should be free to take reasoned exception to the data or views offered in any course of study and to reserve judgment about matters of opinion, but they are responsible for learning the content of any course of study for which they are enrolled. Students who believe that an academic evaluation reflects prejudiced or capricious consideration of student opinions or conduct unrelated to academic standards should contact the dean of the college which offers the class to initiate a review of the evaluation.

---