

# CS320/520N: Organization of Programming Languages Winter 2012

**Class Meetings:** Tue, Thu 3:10pm–5:00pm, Grover E218

**Instructor:** Razvan Bunescu

**Office:** Stocker 337

**Office Hours:** Tue, Thu 11:00am–1:00pm, or by email appointment

**Email:** bunescu @ ohio edu

**Grader:** Jincheng Chen (jc963109 @ ohio edu)

**Class Website:** <http://ace.cs.ohio.edu/~razvan/courses/cs320>

## **Prerequisites:**

- CS240B (Introduction to Computer Science)
- CS300 (Discrete Structures)

## **Textbook:**

*Concepts of Programming Languages (9th edition)*

by Robert W. Sebesta. Addison-Wesley, 2009

## **Recommended Supplementary Text:**

*Compilers: Principles, Techniques, & Tools (2nd edition)*

by Aho, Lam, Sethi, and Ullman. Addison-Wesley, 2007

## **Course Description:**

This course introduces the fundamental syntactic and semantic concepts underlying modern programming languages. The imperative, functional and logic programming paradigms will be discussed, with illustrative examples in C/C++, Java, Ada, Scheme, Python and Prolog. Major topics include:

- Context Free Grammars, Lexical Analysis and Syntactic Parsing.
- Bindings, Type Checking and Scopes.
- Expressions, Control Structures and Functions.

## **Course Outcomes:**

*A: Ability to apply knowledge of Computing and Mathematics appropriate to the discipline.*

Students will be able to:

1. Map evaluation criteria such as cost and readability to particular features of a programming language.
2. Describe and compare the three main programming language paradigms: imperative, functional, and logic.
3. Transform an iterative implementation of a simple function into a recursive version, and vice versa.
4. Given a simple language, design a regular expression, regular grammar or context free grammar that generates it.

5. Use static and dynamic scoping rules to disambiguate variable names in an arbitrary program.
6. Comparatively evaluate static typing vs. dynamic typing.
7. Determine whether particular expressions in a known programming language are referentially transparent.
8. Identify the type coercions that occur in expressions in a known programming language.
9. Identify the parameter passing methods used in a particular C++, Ada, Python, or Scheme program.

*C: Ability to design, implement, and evaluate a computer-based system, process, component or program to meet desired needs.* Students will be able to:

1. Implement a recursive descent parser for a simple arithmetic expression grammar.
2. Implement generators in Python.
3. Use slice notation, lambda expressions, functionals, and list comprehensions in Python.
4. Implement simple functionals such as filter and reduce in Scheme.
5. Implement basic list predicates such as sum and member in Prolog.
6. Implement the factorial function in all the three major programming paradigms.

### **Grading**

30%: Assignments

40%: 2 Midterm Exams

30%: Final Exam

### **Grading Scale:**

A (> 92%) A-(> 90%) B+(> 87%) B(> 83%) B-(> 80%)

C+(> 77%) C(> 73%) C-(> 70%) D+(> 67%) D(> 63%) D-(> 60%)

### **Exam Dates:**

**Midterm:** Thursday, Jan 26, in class (tentative)

**Midterm:** Thursday, Feb 16, in class (tentative)

**Final:** Monday, Mar 12, 2:30pm - 4:30pm

### **Other Important Dates:**

Tuesday, Jan 17: Last day to add class.

Monday, Feb 6: Last day to drop class.

Thursday, Mar 8: Last day of this class.

### **Course and Attendance policies:**

Late assignments: All homework assignments are due before the class. No late submissions will be accepted without prior approval.

Attendance: It is in your best interest to attend all the lectures. Some of the material will not be found in the textbook or on the slides. Extra credit will be awarded for class activity. Also, be sure to check the course website for important announcements on a regular basis.

**Academic Dishonesty Policy:**

All work must be the student's own. All external references used in reports must be properly cited. No credit will be given for duplicate or plagiarized work. Stronger measures, within the guidelines of the Student Handbook, may be taken when conditions warrant. The OU Student Code of Conduct Policy is available online at:

[http://www.ohio.edu/judiciaries/conduct\\_policy.cfm](http://www.ohio.edu/judiciaries/conduct_policy.cfm)

**Other Policies:**

Be sure to notify the professor of any exam conflicts or other extenuating circumstances well in advance. No missed exams will be made up without prior approval.