CS 3200/5200N: Organization of Programming Languages
Spring 2015

Class Meetings: Tue, Thu 12:00pm–1:20pm, Walter Hall 245
Instructor: Razvan Bunescu
Office: Stocker 341
Office Hours: Tue, Thu 10:30–11:30am, or by email appointment
Email: bunescu @ ohio edu
Teaching Assistant: Sadegh Mirshekarian (sm774113@ohio.edu @ ohio edu)
Office: Stocker 285
Office Hours: Mon, Wed 9:00–10:30am, or by email appointment
Class Website: http://ace.cs.ohio.edu/~razvan/courses/cs3200

Prerequisites:
CS 2650 and (CS 3000 or MATH 3050) and C or better in CS 2401

Textbook:
Programming Language Pragmatics (3rd edition)
by Michael L. Scott. Elsevier/Morgan Kaufmann, 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%: 8 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, Feb 25, in class
Midterm: Thursday, Apr 7, in class
Final: Thursday, April 28, 10:10am - 12:10pm

Other Important Dates:

Friday, Jan 22: Last day to add class.
Tuesday, Mar 1: Spring Break, no class.
Thursday, Mar 3: Spring Break, no class.
Friday, Mar 25: Last day to drop class.
Thursday, Apr 21: Last day of this class.

Course and Attendance policies:

Assignments: All homework assignments are due before the class. **No late submissions will be accepted**.

Attendance: It is in your best interest to attend 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 your OU email 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. Additional measures may be imposed by the University Judiciaries, when conditions warrant. Students may appeal academic sanctions through the grade appeal process. The OU Student Code of Conduct Policy is available online at:

`http://www.ohio.edu/communitystandards/academic/students.cfm`

Disability-based Accommodation:

Any student who suspects s/he may need an accommodation based on the impact of a disability should contact the class instructor privately to discuss the students specific needs and provide written documentation from the Office of Student Accessibility Services. If the student is not yet registered as a student with a disability, s/he should contact the Office of Student Accessibility Services.

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. Medical excuse forms need to explicitly mention that the student could not have attended the exam at the specified time due to health concerns.