

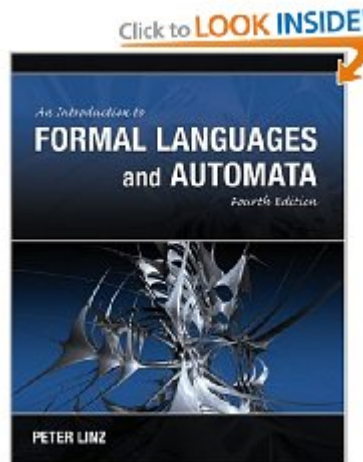
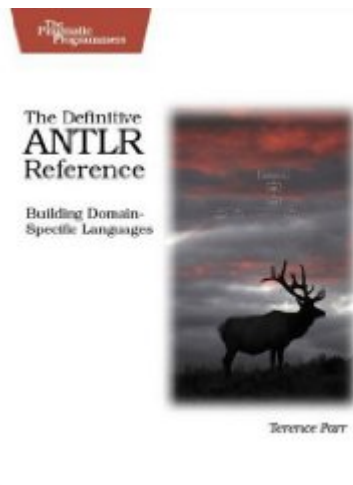
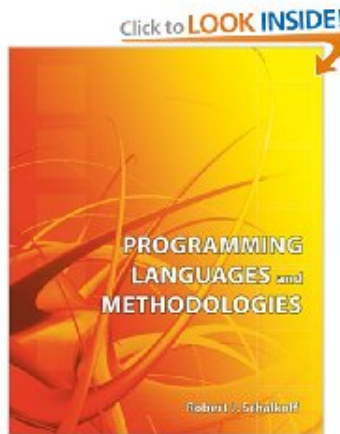
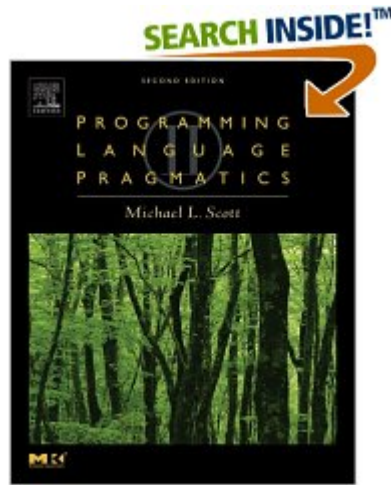
CS 300: Programming Languages

Marshall University, Spring 2009

January 13, 2009

Contents

| | |
|---|----------|
| 1 Course description | 3 |
| 2 Prerequisites | 3 |
| 3 Instructor information | 3 |
| 4 Course schedule and office hours | 3 |
| 5 Instructional materials | 3 |
| 6 Course topics at a glance | 4 |
| 7 Course goals | 5 |
| 8 Measurable student learning outcomes | 6 |
| 9 Course assessment | 7 |
| 9.1 Programming assignments (30%) | 7 |
| 9.2 Pop quizzes (10%) | 7 |
| 9.3 Two midterm exams (40%) | 7 |
| 9.4 Final exam (20%) | 8 |
| 10 muOnline/Blackboard | 8 |
| 11 Policy for students with disabilities | 8 |



1 Course description

In this course, you will learn the structure and features of high-level programming languages and understand how they have been implemented in several actual programming languages. Emphasis is not on learning a particular language (or set of languages) in great detail, but on the process of learning how to learn new programming languages. However, you will write simple programs using C/C++, PROLOG, Mathematica, and MATLAB.

2 Prerequisites

- Intermediate-level proficiency in programming using an object-oriented language such as Java or C++.
- CS210 or equivalent with a grade of C or better.

3 Instructor information

- Dr. V.N. Gudivada, Gullickson Hall Room 205A, Phone: 304-696-5452, Email: gudivada@marshall.edu. Please use muOnline/Blackboard email for course related inquiries.

4 Course schedule and office hours

- This course meets on TuTh 11.00 AM - 12.15 PM in GH 206A.
- Office hours
 - Tuesdays and Thursdays: 2.00 PM - 3.30 PM.
 - Wednesday: 2.00 PM - 5.00 PM.
 - Other times by appointment.

5 Instructional materials

- *Required Textbook:*

[1] Michael L. Scott. *Programming Language Pragmatics*. Morgan Kaufmann, second edition, 2006. ISBN: 0-12-633951-1.

○ *Reference Textbooks (no need to buy):*

- [2] Peter Linz. *An Introduction to Formal Language and Automata*. Jones & Bartlett Pub, fourth edition, 2006. ISBN: 0763737984.
- [3] Terence Parr. *The Definitive ANTLR Reference: Building Domain-Specific Languages*. Pragmatic Bookshelf, 2007. ISBN: 0978739256.
- [4] Susan H. Rodger and Thomas W. Finley. *JFLAP: An Interactive Formal Languages and Automata Package*. Jones & Bartlett Pub, 2006. ISBN: 0763738344.
- [5] Robert J. Schalkoff. *Programming Languages And Methodologies*. Jones & Bartlett Pub, first edition, 2006. ISBN: 0763740594.

○ Web resources

- Popularity of programming languages (Tiobe index)
<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>.
- Comparison of static code checking tools
(<http://www.tiobe.com/index.php/content/paperinfo/SurveyNewGenerationCodeCheckers.h>).
- Programming language coding standards
(<http://www.tiobe.com/index.php/content/paperinfo/HowToUseCodingStandards.html>).
- C programming language Quiz Guide
(http://www.sis.pitt.edu/paws/system_quizguide.htm).
- Ontology of C programming domain
(http://www.sis.pitt.edu/paws/ont/c_programming.rdfs).
- Additional resources: Course notes, other handouts, URLs for additional Web resources will be available on the muOnline/Blackboard system.

6 Course topics at a glance

- A survey contemporary programming languages and their features
- Syntax and semantics
- Binding, scope rules, and type safety
- Abstract data types, algebraic types, and type inheritance and inference
- Expressions and operators

- Control structures
- Method/subprogram implementation
- Concurrency control
- Exception handling
- Generic programming
- Functional programming
- Object-oriented programming

7 Course goals

- Describe the syntax and semantics of contemporary programming languages.
- Discuss how build to build translators and interpreters for domain-specific languages.
- Explain fundamental programming language concepts including expressions, operators, operator precedence and associativity, expression evaluation, fundamental control structures, binding times, scope rules, abstract data types, type inheritance and type safety.
- Compare and contrast various programming languages in terms of their structure, and features; determine their suitability in an application context.
- Describe language runtime support for method implementation, invocation, and return; and discuss parameter passing techniques.
- Explain concurrency concepts and describe language support for concurrent programming.
- Explain exception handling concepts and language support for processing exceptions.
- Compare and contrast procedural, object-oriented, generic, functional, and declarative programming paradigms.
- Write simple but substantial programs using languages based various programming paradigms: C/C++ and MATLAB.

8 Measurable student learning outcomes

A high course grade in CS 300: Programming Languages requires that the student demonstrate most or all of the following:

- **Explains** the technical and pragmatic reasons for the proliferation of programming languages.
- **Articulates** the distinguishing characteristics different programming paradigms and language constructs required for supporting them.
- **Clearly distinguishes** between programming language syntax and semantics, specification schemes for specifying them, and their role in learning new languages and impact on programming productivity and software maintenance.
- **Understands** the issues related to early and late binding, how scope rules govern the visibility of variables and program units, and issues in type safety and type casting.
- **Understands** the notions and applications of abstract data types, algebraic types, and type inheritance and inference; compares and contrasts them.
- **Demonstrates** knowledge of techniques for expression evaluation, and role of operator precedence and associativity in expression evaluation.
- **Understands** the nuances in the variations of the fundamental control structures and is **knowledgeable** in choosing a right control structure for a given context.
- **Demonstrates** an understanding of the need for and issues in concurrency control and **programmatically illustrates** concurrent access to shared resources.
- **Demonstrates** an understanding of the need for and issues in runtime exceptions and **programmatically illustrates** exception raising and exception processing.
- **Articulates** the functions of language runtime system and explains how it: determines visibility of variables and program units; implements method invocation and return; performs memory management; and supports concurrency and exception handling.

- **Recognizes** the need for generic programming, **gained** basic understanding of developing generic software components.
- **Understands** the process for building interpreters and translators for domain-specific languages.
Successfully developed non-trivial programs in C/C++, Mathematica, PROLOG, and MATLAB.

9 Course assessment

The course assessment components include: programming assignments (30%), pop quizzes (10%), two midterm exams (40%), and a final exam (20%). Maximum possible score is 100. Course grade is awarded based on the following scheme:

| <i>Score</i> | <i>Letter Grade</i> |
|-----------------------|---------------------|
| ≥ 90 | A |
| $\geq 80 \ \& \ < 90$ | B |
| $\geq 70 \ \& \ < 80$ | C |
| $\geq 60 \ \& \ < 70$ | D |
| < 60 | F |

9.1 Programming assignments (30%)

There will be several programming assignments spanning multiple languages: C/C++, MATLAB, Mathematica, and PROLOG. Programming assignments will account for 30% of the course grade.

9.2 Pop quizzes (10%)

There will be several pop quizzes (about 10) on the topics discussed in the class and reading assignments. All the quizzes will be administered using muOnline/Blackboard and start precisely at 11.00 AM and end sharply at 11.10 AM. Pop quizzes will account for 10% of the course grade.

9.3 Two midterm exams (40%)

There will be two midterm exams, and each weighs 20% of the course grade. The first midterm exam will be held on [12 February 2009 \(Thursday\)](#), and the second on [19 March 2009 \(Thursday\)](#).

9.4 Final exam (20%)

The [final exam](#) will be a comprehensive one — includes topics those that we have studied during the entire semester. It will be held in [GH 206A on 7 May 2009 \(Thursday\), 10.15 AM - 12.15 AM](#).

10 muOnline/Blackboard

It is important to visit muOnline/Blackboard regularly for up-to-date information about the course. It hosts all the course materials including assignments, handouts, lecture notes, and reading materials. Also, you will use the Blackboard for submitting your team project.

11 Policy for students with disabilities

Marshall University is committed to equal opportunity in education for all students, including those with physical, learning and psychological disabilities. University policy states that it is the responsibility of students with disabilities to contact the Office of Disabled Student Services (DSS) in Prichard Hall 117, phone 304 696-2271, to provide documentation of their disability. Following this, the DSS Coordinator will send a letter to each of the student's instructors outlining the academic accommodation he/she will need to ensure equality in classroom experiences, outside assignment, testing and grading. The instructor and student will meet to discuss how the accommodation(s) requested will be provided. For more information, please visit <http://www.marshall.edu/disabled> or contact Disabled Student Services Office at Prichard Hall 11, phone 304-696-2271.