CS 300: Programming Languages
Marshall University, Spring 2006
Course Syllabus

Contents

1 Course Description **3**

2 Prerequisites **3**

3 Course Goals **3**

4 Class Meeting Time and Location **4**

5 Instructor Information **4**

6 Schedule of Teaching and Learning Activities **4**

Week 1: 10, 12 - Jan - 2006 4
Week 2: 17, 19 - Jan - 2006 5
Week 3: 24, 26 - Jan - 2006 5
Week 4: 31 Jan, 2 Feb - 2006 6
Week 5: 7, 9 - Feb - 2006 6
Week 6: 14, 16 - Feb - 2006 6
Week 7: 21, 23 - Feb - 2006 7
Week 8: 28 Feb, 2 Mar - 2006 7
Week 9: 7, 9 - Mar - 2006 7
Week 10: 14, 16 - Mar - 2006 8
Week 11: 28, 30 - Mar - 2006 8
Week 12: 4, 6 - Mar - 2006 8
Week 13: 11, 13 - April - 2006 9
Week 14: 18, 20 - April - 2006 9
Week 15: 25, 27 - April - 2006 9

7 Textbook **9**

8 Course Assessment and Grading Criteria **10**

9 Team Project **10**

10 Plagiarism and Attendance Policies **11**

11 Due Dates for Assigned Work **12**

12 Teaching and Learning Resources **12**

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 the following languages: MatLab, Mathematica, Python, C/C++, ML/Haskel, and Godel.

This is a fast moving course — demands a minimum of nine hours per week of outside class time. The course employs an assortment of active learning techniques in lieu of traditional classroom lectures. Out-of-class activities form the basis for upcoming class meetings. Therefore, it is essential that students complete the assigned out-of-class activities before coming to the class.

2 Prerequisites

- Intermediate-level proficiency in programming using an object-oriented language such as Java or C++.
- Access to WebCT Vista from your home will be quite helpful.

3 Course Goals

The course has the following primary goals:

1. describe the syntax and semantics of contemporary programming languages,
2. explain fundamental 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.
3. compare and contrast various programming languages in terms of their structure, and features; determine their suitability in an application context,
4. describe language runtime support for method implementation, invocation, and return; and discuss parameter passing techniques.

5. explain concurrency concepts and describe language support for concurrent programming.
6. explain exception handling concepts and language support for processing exceptions.
7. compare and contrast procedural, object-oriented, generic, functional, and declarative programming paradigms.
8. write simple programs using languages based various programming paradigms.

4 Class Meeting Time and Location

9.30 - 10.45 AM, Tuesdays and Thursdays, GH 206A.

5 Instructor Information

Name: Dr. Venkat N. Gudivada, Associate Professor, Engineering & Computer Science.

Phone and email: 304-696-5452; gudivada@marshall.edu.

Office Location: Gullickson Hall, Room 205A.

Office Hours: 11.00 AM – 2.00 PM on Tuesdays and Thursdays. Other times by appointment.

6 Schedule of Teaching and Learning Activities

Week 1: 11, 12 - Jan - 2006

In-Class Activities

- Discussion of syllabus and overview of course activities.
- Overview and discussion of chapters 1 and 2 (textbook).

- In-class demo of Matlab — a scientific and technical computing environment.
- Next out-of-class activities walk-through.

Out-of-Class Activities

- Study chapters 1 and 2 (textbook) and be prepared for discussion.
- Go through Matlab tutorials (one or more of the following): Union College; University of Michigan; University of Utah
- Complete Matlab homeplay on matrices. Refer to the homeplay handout.

Week 2: 17, 19 - Jan - 2006

In-Class Activities

- Guided discussion and reflection on the last out-of-class activities including Matlab Homeplay.
- Discussion on chapters 2 and 3.
- Next out-of-class activities walk-through.

Out-of-Class Activities

- Study chapters 2 and 3.
- Work on Homeplay No 2 (Problems 1.1, 1.2, 1.7, 1.8, and 1.17). Due by end of day on 24 Jan 2006.

Week 3: 24, 26 - Jan - 2006

In-Class Activities

- Guided discussion and reflection on the last out-of-class activities including Homeplay.
- Discuss syntax and semantics (chapters 4 and 5)
- Next out-of-class activities walk-through.

Out-of-Class Activities

- Study Matlab programming model.
- Study syntax and semantics (chapters 4 and 5)

Week 4: 31 Jan, 2 Feb - 2006

In-Class Activities

- Guided discussion and reflection on the last out-of-class activities including Homeplay.
- Continue discussion on syntax and semantics (chapters 4 and 5)
- Next out-of-class activities walk-through.

Out-of-Class Activities

- Work on the homeplay.
- Preview C and C++ material.

Week 5: 7, 9 - Feb - 2006

In-Class Activities

- Guided discussion and reflection on the last out-of-class activities including Homeplay.
- Next out-of-class activities walk-through.

Out-of-Class Activities

-

Week 6: 14, 16 - Feb - 2006

In-Class Activities

- Guided discussion and reflection on the last out-of-class activities including Homeplay.

- Next out-of-class activities walk-through.

Out-of-Class Activities

-

Week 7: 21, 23 - Feb - 2006

In-Class Activities

- Guided discussion and reflection on the last out-of-class activities including Homeplay.
- Next out-of-class activities walk-through.

Out-of-Class Activities

-

Week 8: 28 Feb, 2 Mar - 2006

In-Class Activities

- Guided discussion and reflection on the last out-of-class activities including Homeplay.
- Next out-of-class activities walk-through.

Out-of-Class Activities

-

Week 9: 7, 9 - Mar - 2006

In-Class Activities

- Guided discussion and reflection on the last out-of-class activities including Homeplay.
- Next out-of-class activities walk-through.

Out-of-Class Activities

-

Week 10: 14, 16 - Mar - 2006

In-Class Activities

- Guided discussion and reflection on the last out-of-class activities including Homeplay.
- Next out-of-class activities walk-through.

Out-of-Class Activities

-

Week 11: 28, 30 - Mar - 2006

In-Class Activities

- Guided discussion and reflection on the last out-of-class activities including Homeplay.
- next out-of-class activities walk-through.

Out-of-Class Activities

-

Week 12: 4, 6 - Mar - 2006

In-Class Activities

- Guided discussion and reflection on the last out-of-class activities.
- Next out-of-class activities walk-through.

Out-of-Class Activities

-

Week 13: 11, 13 - April - 2006

In-Class Activities

- Guided discussion and reflection on the last out-of-class activities.
- Next out-of-class activities walk-through.

Out-of-Class Activities

-

Week 14: 18, 20 - April - 2006

In-Class Activities

-

Out-of-Class Activities

-

Week 15: 25, 27 - April - 2006

In-Class Activities

- Team project presentations
- Answer final exam

7 Textbook

The following textbook is required. In addition, the course will substantially draw upon the teaching and learning resources which will be made available on WebCT Vista.

- Kenneth C. Louden. Programming Languages (Second Edition). Thompson (Brooks/Cole), 2003. ISBN: 0-534-95341-7.
- Textbook Author's Web Site

8 Course Assessment and Grading Criteria

The course assessment is based on the following measures:

- Homeplays (Out-of-class Focused Problem Solving): 30%
- Team Project (Formal Written Work): 20%
- Individual Student Conference with the Instructor (should occur by 30-Mar-2006): 5%
- Midterm Exam: 10%
- Final Exam: 10%
- Programming Projects: 5 @ 5% = 25%

Assignment of Letter Grade

Score	*Letter Grade*	Remarks
$>= 90$	A	Achievement of distinction
$>= 80$ & < 90	B	Competent and professional work
$>= 70$ & < 80	C	Below average performance
$>= 60$ & < 70	D	Patently substandard work
< 60	F	Unsatisfactory work

Note that *A* grades are awarded only to those students who have demonstrated distinctive performance in the course.

9 Team Project

You are required to successfully complete a team project. The latter is characterized by:

- Students work on focused but *open-ended tasks*. These are primarily programming language evaluation tasks, which need to be approached from the perspective of different perspectives — readability, writability, type safety,

binding times, extensibility, control abstractions, data abstractions, programming paradigm, parameter passing mechanisms, exception handling, and support for threading.

Students should recognize that there are multiple ways to solve an open-ended problem. Selecting a solution involves judicial analysis of the trade-offs among a set of competing and mutually opposing design and operational parameters.

- Student teams typically consist of two to three members. Team projects are not intended to achieve an outcome where only one student does most or all of the work and the other students simply put their names on the finished product.
- Team projects are designed: to *create positive interdependence* among the team members; to *promote each others learning* face-to-face; to hold each other *accountable for her fair share of the work*; to help *experience the issues that arise in team situations* and to resolve them amicably.
- Constant communication among the team members, and frequent face-to-face meetings of the team are critical to the successful completion of team projects.
- Since team projects are not trivial, they require the use of analysis, design, and project management skills.

Detailed project description including rubrics and submission date will be provided in a separate handout.

10 Plagiarism and Attendance Policies

Plagiarism — submitting somebody else's work in part or whole as one's own work — is a form of academic dishonesty. The procedures and policies as outlined in pages 55 – 64 of the Marshall University Online Graduate Catalog (Spring 2005 Edition) will be applied in dealing with academic dishonesty. Students are strongly urged to become acquainted with these policies and procedures.

It is important that you attend all the classes, especially given that the class meets only once a week. If you miss a class for a valid reason, you are still responsible for completing the in-class and out-of-class activities.

11 Due Dates for Assigned Work

No late submissions will be accepted. Due dates for various assigned work are indicated either on the handouts or WebCT Vista.

12 Teaching and Learning Resources

Visit WebCT Vista.