

# CSC 7101

## Programming Language Structures

### Spring 2009

## Syllabus

Gerald Baumgartner

### Course Summary

The course listing says:

Advanced study of data specification, storage management, and control in programming languages; includes coverage of formal specification languages; languages for concurrent processing; languages that support program verification techniques; and in-depth study of applicative languages.

In addition to these topics, we will cover logic programming and type systems.

### Prerequisites

- CSC 4101: Programming Languages.

### Office Hours

| Who                | Where             | Phone    | E-Mail   | When           |
|--------------------|-------------------|----------|----------|----------------|
| Gerald Baumgartner | Coates 290 or 173 | 578-2191 | gb       | MW 2:00-3:30pm |
| Ajay Panyala       | Coates 173        |          | apanyala |                |

Other office hours by appointment (recommended).

### Important Dates

- Martin Luther King Day: Jan 19
- Mardi Gras: Feb 23-25
- Midterm: Wed, Mar 4
- Spring break: Apr 6-12
- Final: Wed, May 6, 12:30-2:30pm (tentative)

Both exams are comprehensive.

## Reading

There is no required textbook! There is no ideal single textbook out there in print. We will use material from several different books. The copies of the slides I'll hand out should contain enough detail.

- Kenneth Slonneger, Barry L. Kurtz, *Formal Syntax and Semantics of Programming Languages*, Addison-Wesley, Reading, MA, 1995. Available online.
- Frank G. Pagan, *Formal Specification of Programming Languages: A Panoramic Primer*, Prentice-Hall, Englewood Cliffs, NJ, 1981.
- Glynn Winskel, *The Formal Semantics of Programming Languages: An Introduction*, The MIT Press, Cambridge, MA, 1993.
- Ryan Stansifer, *The Study of Programming Languages*, Prentice-Hall, 1995.
- Mailing list `csc7101@csc.lsu.edu`.
- Web page `http://www.csc.lsu.edu/~gb/csc7101/`.

## Homeworks

There will be five or six homework assignments, which will be due at the beginning of class on the due date. A penalty of 15% will be assessed for each day a homework is late up to a maximum of 30%.

## Labs

There will be one programming assignment, which will be due at midnight (11:59pm) of the due date. A penalty of 10% will be assessed for each day the lab is late up to a maximum of 30% after which the lab will not be accepted.

## Grading

|          |     |
|----------|-----|
| Homework | 30% |
| Lab      | 20% |
| Midterm  | 20% |
| Final    | 30% |

## Topics

The following list indicates roughly how much time we will spend on each topic:

| No. of Weeks | Topic                             |
|--------------|-----------------------------------|
| 2            | Attribute Grammars                |
| 3            | Axiomatic Semantics               |
| 1            | Functional Programming            |
| 1            | Review and Midterm                |
| 2            | Logic Programming                 |
| 2            | Operational Semantics             |
| 2            | Type Systems                      |
| 2            | Denotational Semantics and Review |

There is also some flexibility in shortening some of these topics and adding other advanced topics.

## Course Policy

### Grading

The entire course will be graded on a curve. I expect the average grade to be a B. For this reason, I will deduct points rather liberally and I will encourage the grader to do the same. Don't be *too* upset if you don't get what you consider to be a high score. When grading on a curve the absolute score is not that important. To give you a feeling about where you are standing in class, statistics about the scores will be provided periodically.

It is course policy that whoever graded something will be responsible for handling grading disputes. I will grade the midterm exam and the final exam. The grader will grade the homeworks and the labs. Grades become final one week after a homework or exam is handed back. This should leave ample time to resolve grading disputes.

### Homework Standards

All written work submitted must carry the student's name and must be reasonably neat and well organized. Any work that cannot be read easily will score zero points. A reasonable standard of English expression and grammar is also required. The same requirements apply to exams.

### Programming Standards

The algorithm used must be essentially correct. Obviously, the program should (compile and) run. Because of the complexity of the programs, no credit can be given for a program that doesn't run. If a program dumps core, only partial credit will be given.

### Honesty

I will treat you as professionals, and you should plan on conducting yourself as such. This course presents many important concepts you will need throughout your career as a computing professional, so it is important that *each student* do *all* the assignments and projects and learn the material.

You are free to discuss homework assignments and labs with others. However, the solutions you submit are to be developed by yourself. *Cheating is a very serious offense and will not be tolerated.* Supplying others with material is also against this rule. The policy is that the supplier and receiver of information will both be punished.