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

<table>
<thead>
<tr>
<th>Who</th>
<th>Where</th>
<th>Phone</th>
<th>E-Mail</th>
<th>When</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gerald Baumgartner</td>
<td>Hatcher 352</td>
<td>578-2191</td>
<td>gb</td>
<td>MW 10:00–11:30am</td>
</tr>
<tr>
<td>TBA</td>
<td></td>
<td></td>
<td></td>
<td>TBA</td>
</tr>
</tbody>
</table>

Other office hours by appointment (recommended).

Important Dates
- Martin Luther King Day: Jan 16
- Mardi Gras: Feb 27
- Midterm: Mon, Mar 6
- Spring break: Apr 8–16
- Final: Thu, May 4, 12:30–2:30pm

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.

- Mailing list csc7101@csc.lsu.edu.
- Web page http://www.csc.lsu.edu/~gb/csc7101/.

Homeworks

There will be four or five 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%.

Projects

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

Grading

| Homeworks | 20% |
| Projects  | 30% |
| Midterm   | 20% |
| Final     | 30% |

Topics

I will be restructuring the course this semester. Because of this I don’t know yet exactly how much time we will spend on which topic.

For reference, the following list indicated roughly how much time was spent on each topic in previous semesters:

<table>
<thead>
<tr>
<th>No. of Weeks</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Attribute Grammars</td>
</tr>
<tr>
<td>3</td>
<td>Axiomatic Semantics</td>
</tr>
<tr>
<td>1</td>
<td>Functional Programming</td>
</tr>
<tr>
<td>1</td>
<td>Review and Midterm</td>
</tr>
<tr>
<td>2</td>
<td>Logic Programming</td>
</tr>
<tr>
<td>2</td>
<td>Operational Semantics</td>
</tr>
<tr>
<td>2</td>
<td>Type Systems</td>
</tr>
<tr>
<td>2</td>
<td>Denotational Semantics and Review</td>
</tr>
</tbody>
</table>
There is also some flexibility in shortening some of these topics and adding other advanced topics.

My plan so far is to swap the first two topics, to add a programming assignment related to these topics, to remove denotational and maybe also operational semantics, and to spend more time on type systems.

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 grader or I may use tools for detecting cheating on programming assignments. The policy is that the supplier and receiver of information will both be reported to the Dean of Students.