CSC 4101
Programming Languages
Fall 2019
Syllabus
Gerald Baumgartner

Course Summary

Principles of programming language design; specification of syntax and semantics; underlying implementation of block structured languages; dynamic memory allocation for strings, lists, and arrays; imperative versus applicative programming; logic programming; modern programming languages.

Prerequisites

- CSC 3102: Advanced Data Structures and Algorithm Analysis.

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>Patrick F. Taylor 3272D</td>
<td>578–2191</td>
<td><a href="mailto:gb@csc.lsu.edu">gb@csc.lsu.edu</a></td>
<td>MW 9:30–11:00am</td>
</tr>
<tr>
<td>Monazil Chowdhury</td>
<td></td>
<td></td>
<td><a href="mailto:mchow15@lsu.edu">mchow15@lsu.edu</a></td>
<td></td>
</tr>
</tbody>
</table>

Other office hours by appointment.

Reading

- CSC 4101 Mailing List csc_4101@eng.lsu.edu.
Important Dates

- Labor Day: Sep 2
- Midterm Exam: Mon, Oct 14, 12:00–1:20pm
- Turkey Day Break: Wed, Nov 27
- Final Exam: Thu, Dec 12, 7:30–9:30am

Both exams are comprehensive.

Programming Assignments

There will be three programming assignments, which will be due at midnight (11:59pm) of the due date. The first two programming assignments will be in an object-oriented language TBA (Java, C++, C#, and/or Python). The third programming assignment will be in Scheme. The first two programming assignments will be in teams of two, the third will be done individually. Since all program assignments are submitted and graded on a Linux server, submissions must be compatible with the language versions installed there. There will also be small programming problems, such as 1–10 liners in Scheme, ML or F#, and Prolog as parts of homeworks.

A penalty of 10% will be assessed for each day a project is late up to a maximum of 30% after which it will not be accepted. The final project may not be turned in later than Saturday midnight after the last day of classes.

Homeworks

There will be five homework assignments, which will be due at the beginning of class on the due date. A penalty of 20% will be assessed if a homework is submitted at the beginning of the next class period. Homeworks later than that will not be accepted.

Annotated Bibliography

Honors students who want to honorize this course as well as graduate students also must submit an annotated bibliography on a programming language research topic.

Grading

The following weights will be used for grading. The homeworks all weigh equally. The first two projects are worth 13% each, while the third project is worth 10%.

<table>
<thead>
<tr>
<th>Component</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Projects</td>
<td>36%</td>
</tr>
<tr>
<td>Homeworks</td>
<td>15%</td>
</tr>
<tr>
<td>Bibliography</td>
<td>3%</td>
</tr>
<tr>
<td>Midterm</td>
<td>24%</td>
</tr>
<tr>
<td>Final</td>
<td>25%</td>
</tr>
</tbody>
</table>

For honors and graduate students the maximum score with the bibliography assignment is 103 points. For computing the letter grades, that total score is then divided by 1.03.

The course will be graded partly on a curve. 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 an idea of where you are standing in class, statistics about the scores will be provided periodically.
In the past, the letter grade cutoffs have been approximately in 15 point increments. Depending on the difficulty of the exams and the strictness of grading, the cutoffs might be moved up or down a little. After that, each letter grade range will be divided into three parts for +/- grades.

**Topics**

The following list of topics is the ordering of subjects covered in this course. The time spent on each subject will vary. I don’t know yet when the homework and project due dates will be. So the following list of topics is only a rough outline.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction, Functional Programming</td>
<td></td>
</tr>
<tr>
<td>Scheme, OO Programming</td>
<td>HW 1</td>
</tr>
<tr>
<td>Syntax and Parsing, OO Programming</td>
<td>HW 2</td>
</tr>
<tr>
<td>OO Programming, Data Types</td>
<td>Project 1</td>
</tr>
<tr>
<td>OO Programming, Types, Scoping</td>
<td>HW 3</td>
</tr>
<tr>
<td>Subprograms and Parameter Passing</td>
<td>Midterm</td>
</tr>
<tr>
<td>Parameter Passing, Subprogram Impl.</td>
<td>HW 4</td>
</tr>
<tr>
<td>Subprogram Impl., Exception Handling</td>
<td>Project 2</td>
</tr>
<tr>
<td>Functional Programming, ML or F#</td>
<td>HW 5</td>
</tr>
<tr>
<td>Logic Prog., Prolog, Scripting Lang.</td>
<td>Project 3</td>
</tr>
</tbody>
</table>

**Due Dates and Grading**

The time allotted for each homework assignment will be made quite generous, as such, the penalty for turning in late is high: 20% for turning it in in the next class period. Homeworks need to be submitted as paper copies, not electronically.

Since the time needed for finishing a programming assignment is harder to estimate and to allow fixing severe bugs that show up close to the deadline, programming assignments can be submitted up to three days after the official deadline. For each day past the deadline, a penalty of 10 percent will be incurred. Programming assignments will be submitted electronically. Projects will be due at midnight.

Without prior arrangements in case of extenuating circumstances, submission of homeworks and projects past the late deadline is *not* allowed, such work will *not* be graded, and you will receive no credit. It is your responsibility to make sure that you have completed your work with enough time to submit your materials.

Grading disputes can be submitted in writing with accompanying documentation, or in person during regular office hours. It is course policy that whoever graded the work will be responsible for handling disputes. In general, I will grade the midterm exam and the final exam. The graders will grade the homeworks and the programming assignments. Grades become final one week after a homework, project, or exam is handed back. This should leave ample time for resolving 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 may 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 some of the programs, very little or no credit can be given for a program that doesn’t run. If a program dumps core or throws a runtime exception, only partial credit will be given.

Since programming assignments might build on top of a previous project, it is very important to get each submission to run without runtime exceptions or core dumps and to structure the program so it can be easily extended.

I expect your work to exhibit high standards of programming style and layout, reflecting your expertise as a computer professional. Poor style and documentation may result in points being deducted.

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.

There will be several homework assignments and programming projects. You are free to discuss these assignments with others. However, the programs and homework solutions you submit are to be developed by yourself. Cheating is a very serious offense and will not be tolerated. The graders or I may use tools for detecting cheating on programming assignments. Supplying others with material is also against this rule. The policy is that the supplier and receiver of information will both be punished.

Save all handwritten notes and printouts you generate as you work on a project and keep them until the end of the quarter so as to protect yourself in the event that someone “borrows” your program, or the version you submit is lost.

Computer Account Security and Use

To help others resist the temptation of using your work, you should maintain proper security on your computer account. In particular, keep your password from others and do not alter the protection on any of your files. To give others access to your account or files or printouts of your programs is the same as giving them the information directly and will be dealt with accordingly. Any trouble with computer accounts should be referred to an instructor as soon as possible.

When a program has been submitted electronically, you should maintain an unedited version of what you submitted (with the correct time stamp) until after that program has been graded. It has also been found beneficial to make a copy of a file before editing that file, in case a system crash occurs during the edit session. When typing in a program for the first time, or when making major changes, you should save your work regularly.