Instructor
Amy Sliva, M.S.
Office: AVW 4468
Office Hours: MTTh 11:00am-12:00pm, or by appt.
asliva@umiacs.umd.edu

Teaching Assistant
Bryan Robbins
Office: TA Room, AVW 1112
Office Hours: M 12:00pm-2:00pm, WF 11:00am-1:00pm, or by appt.
brobbins@umd.edu

Course Description
Prerequisites: C or better in CMSC 212 (or CMSC 214), and in CMSC 250
Credits: 3 credits
Lecture: MTuTh 9:30–10:50, CSI 2107
Discussion: WF 9:30–10:50, CSI 2107

A study of programming languages, including their syntax, semantics, and implementation. Several different models of languages are discussed, including procedural (e.g., C, Pascal), functional (e.g., ML, LISP, Scheme), rule-based (e.g., Prolog), and object-oriented (e.g., Java, C++, Smalltalk). Language features such as formal syntax, scoping and binding of variables, higher-order programming, typing and type polymorphism, and object inheritance are explored.

List of Topics (Tentative)
The following list of lecture topics will vary according to the pace of lecture.

- Administrative and course introduction (1 lecture)
- Scripting Languages (Ruby, 2 lectures)
  - Implicit vs. explicit declarations
  - Dynamic vs. static typing
  - Text processing and string manipulation
- Regular expressions and finite automata (3 lectures)
- Context-free grammars (3 lectures)
• Functional programming (OCaml, 5 lectures)
  – Lists and recursion
  – Higher-order programming and closures
  – Types and polymorphism
  – Data types and pattern matching
  – Modularity and abstraction

• Concurrency (5 lectures)
  – Basics of multi-threaded programming
  – Shared memory concurrency
  – Message passing concurrency
  – Design idioms and comparisons

• Environments, scoping, and binding (2 lectures)
  – Functions and procedures
  – Parameter passing mechanisms
  – Dynamic vs. static scoping
  – Run-time implementations

• Advanced Topics (2 lectures)
  – Lambda Calculus
  – Garbage collection and memory management
  – Historical overview

Computing Resources

Programming projects will be developed on the OIT Linuxlab UNIX Cluster, linuxlab.umd.edu. You will use your own OIT account to access the cluster and do coursework, so if you do not have a OIT account, request one immediately online.

If you have access to another system you are welcome to do your development there instead, but all project submissions must work correctly on the Linuxlab cluster, and your projects will be graded solely based on their results on the cluster (by the submit server). Because language and library versions may vary with the installation, in unfortunate circumstances a program might work perfectly on your system but not work at all on the cluster. Thus we strongly recommend that if you develop any project on another system, you should complete it several days early to have time to address any compatibility problems.
Office Hours and Web Forum

Office hours are posted above and on the course website at http://www.cs.umd.edu/class/summer2010/cmsc330.

While we will provide assistance with assignments during office hours, you are responsible for developing and debugging your own programs. Do not rely on the instructional staff to make your project work. Lower-level CMSC courses provide extensive debugging and development help in office hours, but upper-level CMSC courses expect students to complete projects with minimal extra help. Therefore in CMSC 330, we will provide less debugging help than some students may be used to. If you come in with a question, expect to be pointed in the right direction, but it will be up to you to finish solving the problem.

Important announcements will be made in class or in the class web forum. Please make it a habit to check the web page and class web forum daily. You may also use the web forum to ask general questions of interest to the class as a whole, e.g., administrative issues or project clarification questions. Please do not post any information that would violate the university academic integrity policy.

Grading

You are responsible for all material discussed in lecture and discussion section and posted on the class web page, including announcements, deadlines, policies, etc. During the semester we may provide ungraded practice homework exercises and solutions. While we will not collect these exercises, completing them is essential preparation for exams. You may work together on these ungraded homeworks, and you may of course come to office hours for additional help.

Your final course grade will be (tentatively) determined according to the following percentages:

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-6 Programming Projects</td>
<td>50%</td>
</tr>
<tr>
<td>Quizzes</td>
<td>5% total</td>
</tr>
<tr>
<td>2 Midterms</td>
<td>12.5% each</td>
</tr>
<tr>
<td>Final</td>
<td>20%</td>
</tr>
</tbody>
</table>

Any request for reconsideration of any grading on coursework must be submitted within one week of when it is returned. Exam regrading requests must be made in writing. Any coursework submitted for reconsideration may be regraded in its entirety, which could result in a lower score if warranted.

Final course grades will be curved as necessary, based on each student’s total numeric score for all coursework at the end of the semester. Important: Completing the programming assignments is an essential part of the course. Therefore, we may fail any student who does not make a good-faith attempt on all course projects, regardless of the student’s performance or scores on the other coursework.
Programming Projects

Projects must be submitted electronically following the instructions given in class. Projects may not be submitted by any other means (e.g., please do not email your projects to us). It is your responsibility to test your program and verify that it works properly before submitting. All projects are due at 11:59:59pm on the day indicated on the project assignment, according to the submission server’s internal clock.

Projects may be submitted up to 24 hours late for a 10% penalty. For example, a project that would earn 90 points for an on-time submission will earn 81 (which is 90 times 0.90) if submitted late. Note that your project score as it appears on the project submission server will not include any late penalties. Any penalties will be incorporated into the final project grade on the grade server. If you submit both on-time & late, your project will received the maximum of the penalty-adjusted scores.

Project extensions will not be granted due to system problems, network problems, power outages, etc., so do not wait to submit a project until the night it is due. You may submit multiple times up to the deadline, and only your last on-time submission is graded. Similarly, if you submit late, only your last submission before the deadline will be graded. No consideration in grading will be made for errors made in transferring files or submitting the wrong version of your project. Having a working, unsubmitted version will not count; only submitted code will be be counted.

Unlike lower-level programming classes, we will not provide you with test cases before projects are due. Instead, you will be responsible for developing your own techniques for testing your projects. To reiterate: your projects will be graded based on test cases not provided in advance. Because grading is done automatically, you must follow the project specification exactly. Also, while projects will generally not be graded on style or documentation, we reserve the right to manually grade program source code for some projects.

Finally, any ”hard coding” in a project assignment will result in a score of zero for that project, and is considered a bad-faith effort. Hard coding refers to attempting to make a program appear as if it works correctly, when in fact it does not. One example of hard coding would be printing the desired output instead of computing it. This is only one example, and if you have any questions as to what constitutes hard coding, be sure to ask ahead of time.

Exam Scheduling

The class includes two midterms and a final exam. Tentative dates for the exams are:

Midterm 1: June 23
Midterm 2: July 7
Final Exam: July 23

The exact dates will be confirmed later, and may vary due to the progress of lecture and other factors. The midterms and final exam will be held in discussion section, though possibly in a different location. Since all exams will be held during normal class hours, there should be no conflicts. If there is a conflict with any of the three exams, you must inform the instructor within the first two weeks of the class for any allowances to be made.
Excused Absences

Besides the policies in this syllabus, the University’s policies apply during the semester. Various policies that may be relevant appear in the Undergraduate Catalog.

If you experience difficulty during the semester keeping up with the academic demands of your courses, you may consider contacting the Learning Assistance Service in 2201 Shoemaker Building at (301) 314-7693. Their educational counselors can help with time management issues, reading, note-taking, and exam preparation skills.

Missing an exam for reasons such as illness, religious observance, participation in required university activities, or family or personal emergency (such as a serious automobile accident or close relative’s funeral) will be an excused absence. However, excused absences must be requested in writing and must include documentation that the absence qualifies as excused. For example, for medical absences, you must furnish documentation from the health care professional who treated you. The documentation should include the name and phone number of the health care professional and should explicitly state the dates and times that you were incapacitated and therefore unable to attend. The dates and times must include the date of the missed exam; an illness preceding an exam does not constitute an excused absence. Also, simply being seen by a health care professional does not constitute an excused absence. Furthermore, self-documentation of illness is not sufficient support to excuse an absence. Excused absences will not be given unless documentation as described is provided. If you become ill, keep in mind that the University Health Center will not provide medical documentation.

It is the University’s policy to provide accommodations for students with religious observances conflicting with exams, but it is the your responsibility to inform the instructor in advance of intended religious observances. Written notice must be provided immediately upon an exam date being announced or confirmed in order for an absence to be excused. If you have a conflict with one of the planned exams, you must inform us prior to the end of the first two weeks of the class.

For missed exams due to excused absences, the instructor will arrange a makeup exam. However, unless immediate notice is given as early as possible of the reason for any missed coursework, an excused absence may not be granted. If you might miss an exam for any other reason other than those above, you must contact the instructor in advance to discuss the circumstances. We are not obligated to offer a substitute assignment or to provide a makeup exam unless the failure to perform was due to an excused absence.

The policies for excused absences do not apply to project assignments. Projects will be assigned with sufficient time to be completed by students who have a reasonable understanding of the necessary material and begin promptly. In cases of extremely serious documented illness of lengthy duration or other protracted, severe emergency situations, the instructor may consider extensions on project assignments, depending upon the specific circumstances.

Students with Disabilities

Students with disabilities who have been certified by Disability Support Services as needing any type of special accommodations should see the instructor as soon as possible, during
the schedule adjustment period. All arrangements for exam accommodations as a result of disability **must** be made and arranged with the instructor **at least** three business days prior to the exam date, or accommodations **will not** be made.

**Academic Integrity**

The Campus Senate has adopted a policy asking students to include the following statement on each examination or assignment in every course: "I pledge on my honor that I have not given or received any unauthorized assistance on this examination (or assignment)." Consequently, you will be requested to include this pledge on each exam and project. Please also carefully read the Office of Information Technology’s policy regarding acceptable use of computer accounts.

Programming projects are to be written **individually**, therefore cooperation or use of unauthorized materials on projects is a violation of the University’s Code of Academic Integrity. **Any evidence** of this, or of unacceptable use of computer accounts, use of unauthorized materials or cooperation on exams or quizzes, or other possible violations of the Honor Code, **will be submitted** to the Student Honor Council, which could result in an XF for the course, suspension, or expulsion.

- **For learning the course concepts** (including the programming languages), students are welcome to study together or to receive help from anyone else. You may discuss with others the project requirements, the features of the programming languages used, what was discussed in class and in the class web forum, and general syntax errors. Examples of questions that would be allowed are "Does a Java class definition end in a semicolon?" or "What does a 'class not found' error indicate?", because they convey no information about the contents of a project.

- **When it comes to actually writing a project assignment**, other than help from the instructional staff a project must solely and entirely be your own work. Working with another student or individual, or using anyone else’s work **in any way** except as noted in this paragraph, is a violation of the code of academic integrity and will be reported to the Honor Council. You may not discuss design of any part of a project with **anyone** except the instructor or teaching assistants. Examples of questions you may **not** ask others might be "How did you implement this part of the project?" or "Please look at my code and help me find my stupid syntax error!". You may not use any disallowed source of information in creating either their project design or code. When writing projects you are free to use ideas or **short fragments** of code from **published** textbooks or **publicly** available information, but the specific source must be cited in a comment in the relevant section of the program.

**Violations of the Code of Academic Integrity include, but are not limited to:**

1. Failing to do all or any of the work on a project by yourself, other than assistance from the instructional staff.

2. Using any ideas or any part of another person’s project, or copying any other individual’s work in any way.
3. Giving any parts or ideas from your project, including test data, to another student.

4. Allowing any other students access to your program on any computer system.

5. Transferring any part of a project to or from another student or individual by any means, electronic or otherwise.

If you have any question about a particular situation or source then consult with the instructors in advance. Should you have difficulty with a programming assignment you should see the instructional staff in office hours, and not solicit help from anyone else in violation of these rules.

It is the responsibility, under the honor policy, of anyone who suspects an incident of academic dishonesty has occurred to report it to their instructor, or directly to the Honor Council.

Every semester the department has discovered a number of students attempting to cheat on project assignments, in violation of academic integrity requirements. Students’ academic careers have been significantly affected by a decision to cheat. Think about whether you want to join them before contemplating cheating, or before helping a friend to cheat.

You are welcome and encouraged to study and compare or discuss their implementations of the programming projects with any others after they are graded, provided that all of the students in question have received nonzero scores for that project assignment, and if that project will not be extended upon in a later project assignment.

Course Evaluations

If you have a suggestion for improving this class, don’t hesitate to tell the instructor or TAs during the semester. At the end of the semester, please don’t forget to provide your feedback using the campus-wide CourseEvalUM system. Your comments will help make this class better. CourseEvalUM is scheduled to be open between Tuesday, December 1 and Sunday, December 13, but this is subject to change by campus. Right to Change Information

Although every effort has been made to be complete and accurate, unforeseen circumstances arising during the semester could require the adjustment of any material given here. Consequently, given due notice to students, the instructors reserve the right to change any information on this syllabus or in other course materials.