CMSC 330 Organization of Programming Languages Fall 2012

1 Prerequisites and description

Prerequisites: C− or better in CMSC 216 and in CMSC 250
Credits: 3 credits

This course covers the semantics of programming languages and their runtime organization. Different models of languages are discussed, including procedural (e.g., C), functional (e.g., ML, LISP), rule-based (e.g., Prolog), and object–oriented (e.g., C++, Smalltalk). Runtime structures, including dynamic versus static scope rules, storage for strings, arrays, records, and object inheritance are discussed. Language features such as scoping and binding of variables, higher–order programming, typing and type polymorphism, pointers, object inheritance, and exceptions are explored.

2 Contact information

2.1 Email contact

Unfortunately we're not able to explain most course material via email, due to time constraints and other factors. Answering such questions is more appropriate for class discussion or personal communication. Please discuss administrative issues as well as course material in person when possible (office hours, and before and after class are good times), and use email in case of urgent or emergency matters only. Lastly, it is not practical to provide detailed information or assistance regarding programming assignments via email, and attempting to do so often results in students receiving incomplete or inadequate information.

2.2 Instructors

Sections 0101, 0102, 0103

Name:	Larry Herman
Office:	1111 A. V. Williams
Phone:	(301) 405–2762
Email:	larry@cs.umd.edu (∗)

Sections 0201, 0202, 0203, 0204

Name:	Dr. Nick Feamster
Office:	4135 A. V. Williams
Phone:	(301) 405–8010
Email:	feamster@cs.umd.edu (∗)

(∗) See Section 2.1 below regarding email.
Office hours will be provided in a separate handout shortly.

2.3 Teaching assistants

name	duties	email
Tommy Pensyl	teaching, 0101 & 0102	tpensyl@cs.umd.edu (∗)
Tammy Tran	teaching, 0103 & 0201	tnt@cs.umd.edu (∗)
Derek Juba	teaching, 0202 & 0203	juba@cs.umd.edu (∗)
Varun Nagaraja	teaching, 0204, and grading	varun@cs.umd.edu (∗)
Hao Li	grading	haoli@cs.umd.edu (∗)

Office hours will be provided in a separate handout shortly.

While the TAs can provide assistance with programming assignments during office hours, you are responsible for developing and debugging your own programs. You should therefore not rely on the instructional staff for getting a project to 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 you should expect to be pointed in the right direction, but then it will be up to you to finish solving the problem on your own.

3 Resources, class webpage, and Piazza

There are no required or recommended texts for this course this semester. Resources and sources of information for the languages and concepts to be taught will be provided during the semester.

Various course materials will be made available on the class webpage, which can be reached by clicking on following link:

www.cs.umd.edu/class/fall2012/cmsc330

Accessing parts of the webpage will require an ID and password to be provided in class.

Course material or concepts may be discussed in Piazza, an online forum, linked to from the class webpage.

Programming will be done on the OIT Grace Cluster, grace.umd.edu. Students will use their own TerpConnect accounts to access the Grace cluster and do coursework, so students who don't have a TerpConnect account should request one online immediately at www.oit.umd.edu/new. Although programming may be done on other systems that you have access to, it is recommended that you do your work on the Grace cluster. All project submissions must work correctly on those machines, and your projects will be graded solely based on their results on the cluster and the CMSC department submit server. Because language and library versions may vary with the installation, in unfortunate circumstances a program might work perfectly on another system but not work at all on the Grace cluster or the submit server. Thus we strongly recommend that if you want to develop any project on another system, you should complete it **several days early** to have time to address any compatibility problems.

4 Quiz, exam, and final dates

Quizzes will be given in discussion section and will cover discussion and lecture material. They will be announced in a prior class.

Midterm exams will be held during lecture. Their dates will be confirmed later, and could possibly be changed depending on lecture progress and other factors. The expected dates are below.

The final exam date and time will be rescheduled **only** for students having another final at exactly the same time, or for students with more than three final exams scheduled on the same day. (The only courses that students who are enrolled in CMSC 330 should be able to take that have finals at the same time as its final are BIOM 301, EDMS 451, an ENMA 300.) If either of these situations applies to you, you must inform your instructor **at least two weeks in advance** of the final exam time for allowances to be made. Also please inform your instructor immediately if you have a conflict with a scheduled midterm date, or any other important date as the semester progresses.

Exam #1:	Thursday, October 11
Exam #2:	Tuesday, November 13
Final exam:	Thursday, December 13, 4:00–6:00 p.m., location TBA

5 Attendance, homework, and general grading policies

Students are responsible for the material covered, and announcements, deadlines, policies, etc., discussed in lecture and discussion section, regardless of whether they were in class or not. Students are likewise responsible for announcements and information appearing in Piazza. **It's understood that students may occasionally have to miss class for various reasons, but office hours, email, and Piazza are not intended as a replacement for class attendance**.

Various practice (homework) exercises and solutions will be provided during the semester; these will not be collected or graded, but are extremely important for testing your knowledge of the material in preparation for quizzes and exams. You are encouraged to work together on these problems. If you have difficulty solving them, feel free to ask questions in Piazza, or see the instructional staff in office hours as soon as possible.

Coursework will count toward the final grade according to the following percentages:

Midterms:	two midterms	32.5%	(equally weighted)
Final:	will be comprehensive	30%	
Quizzes:	in discussion section	5%	(equally weighted)
Programming projects:	five or six coding assignments	32.5%	

A request for reconsideration of the grading on any coursework must be submitted **within one week** of when it is returned. Exam regrading requests must be made in writing. Coursework submitted for reconsideration may be regraded in its entirety.

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.

Project submission and grading policies will be provided with the first project assignment.

6 Absences and accommodations

Besides the policies in this syllabus, various University policies may apply to students during the semester. Policies that may be relevant appear in the Undergraduate Catalog, at www.umd.edu/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.

6.1 Excused absences

6.1.1 Illness

Any student who needs to be excused for an absence from a single lecture or discussion section for reasons of medical necessity must, according to the University policy:

- Make a reasonable attempt to inform his or her instructor of his or her illness prior to the class.
- Upon returning to the class, present his or her instructor with a self–signed note attesting to the date of their illness. Each note must contain an acknowledgment by the student that the information provided is true and correct. Providing false information to University officials is prohibited under Part 9(h) of the Code of Student Conduct (V–1.00(B) University of Maryland Code of Student Conduct) and may result in disciplinary action.
- This self–documentation may not be used for the major scheduled grading events as defined below and it may only be used only once during the semester.

If a student needs to be excused for a prolonged illness (for this course this means missing two or more consecutive class meetings) or if a major scheduled grading event is missed due to illness, written documentation of the illness from the Health Center or from an outside health care provider must be provided. This must include the contact information of the provider, verify dates of treatment, and indicate the time that the student was incapacitated, in the provider's opinion, and unable to meet academic responsibilities. Diagnostic information need not be given. The major scheduled grading events for this course are the two midterms and the final exam whose dates are given above in Section 4.

6.1.2 Other excused absences

An excused absence will be given for other University–approved reasons, such as religious observance, participation in required university activities, or family or personal emergency, provided that:

- Students requesting an excused absence furnish documentary support of the cause of the absence where feasible.
- The maximum possible advance notice is given.

6.1.3 Projects

The policies for excused absences don't apply to project assignments. These 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 protracted, serious illness, or severe emergency situations, short extensions on projects may be considered, depending upon the circumstances. In such a situation you must contact your instructor as soon as possible.

6.2 How excused absences are handled

All arrangements for excused absences must be made with the instructor (even if the coursework that was missed was in discussion section).

An excused absence for an exam will be handled by giving a makeup exam.

For an excused absence for a single quiz, rather than a makeup the score will be computed at the end of the semester as the average of the student's scores for the other quizzes. Missing more than one quiz will be handled by giving a makeup quiz.

6.3 Students with disabilities

Any student eligible for and requesting reasonable academic accommodations due to a disability is requested to provide, to the instructor in office hours, a letter of accommodation from the Office of Disability Support Services (DSS) within the first two weeks of the semester.

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 cannot be made.

7 Academic integrity

The Campus Senate has adopted a policy asking students to include the following statement on each examination or major 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 exams and projects.

Please carefully read the Office of Information Technology's policy regarding acceptable use of computer accounts and resources at www.nethics.umd.edu/aup.

Unless otherwise noted, all coursework– including programming coursework– is to be done **individually**, so cooperation or use of unauthorized materials on programming assignments 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. Students may discuss with others the project requirements, the features of the programming languages used, what was discussed in class and in Piazza, and **general** syntax errors. Allowable questions are ones that convey no information about the contents of a student's project solution.
- When it comes to actually writing a project assignment, other than help from the instructional staff a project must solely and entirely be a student's 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. Students may not discuss design of any part of a project with **anyone** except the instructors or teaching assistants. Examples of questions which students 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 the stupid syntax error!". Students may not use any disallowed source of information in creating either their project design or code. When writing projects students are free to use ideas or **short fragments** of code from **published** textbooks or **publicly available** information, or information or examples provided by the instructional staff, provided that the specific source is cited in a comment in the relevant section of the program.

VIOLATIONS OF THE CODE OF ACADEMIC INTEGRITY MAY 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 your instructor in advance. Should you have difficulty with a programming assignment you should **see the teaching assistants in office hours**, 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.

Students 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** the project in question will not be extended upon in a later project assignment.

8 Course topics (SUBJECT TO CHANGE)

The following list of lecture topics may vary according to the pace of lecture:

- Administrative and course introduction
- Scripting languages (Ruby)
  - Implicit vs. explicit declarations
  - Dynamic vs. static typing
  - Text processing and string manipulation
  - Data structures in Ruby
- Regular expressions
  - Regular expressions in Ruby
  - Formal definitions
- Finite automata
  - DFAs

  - NFAs
  - Uses and applications
- Environments, scoping, and binding
  - Functions and procedures
  - Parameter passing mechanisms
  - Dynamic vs. static scoping
  - Runtime implementations
- Functional programming (OCaml)
  - Lists and recursion
  - Higher–order programming
  - Types and polymorphism
  - Data types and pattern matching
  - Modules
  - Closures
- Context–free grammars
- Concurrency and multithreading
- Parameters and scope
- Polymorphism and generics
- Functional programming in object–oriented languages
- Programming language theory
  - Lambda calculus
  - Operational semantics
- Pointers and garbage collection
- Historical overview of programming languages

9 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 instructor reserves the right to change any information on this syllabus or in other course materials.

10 Copyright