HON 313: An Honors Tutorial on
Literate Programming and Programming Fundamentals

Sudarshan S. Chawathe

University of Maine

Spring 2008

This course addresses two primary topics: (1) Literate programming, the writing of computer programs that meant primarily for a human reader, their interpretation by a machine being an important, but secondary, consideration. (2) Programming fundamentals, the theoretical and practical foundations of computer programming. The course is suitable for students irrespective of their programming experience: it should suit those with no programming experience as well as those with extensive experience. Further details appear in the introduction below. This course satisfies the university's *Mathematics* and *Writing-Intensive* General Education requirements.

News and Reminders:
- **This material is being updated.**
- Please use the course newsgroup for announcements and discussions.
- Class newsgroup: Local group `umaine.hon313` on NNTP server `news.cs.umaine.edu`. Web interface to get started: `http://cs.umaine.edu/~chaw/news/`.
- Please use the PDF version of this document for printing and reference: `hon313.pdf`

Introduction

The phrase *literate programming* refers to the creation of computer programs that are meant primarily for human readers. Like ordinary computer programs, literate programs can be transformed, using automated tools, into a form suitable for execution on a computer. However, unlike ordinary programs, their primary focus is a human reader. To that end, all aspects of a literate program, such as the problem definition, motivation, possible solutions, high-level and low-level design, and details of the code, are described in a manner that emphasizes a logical flow of ideas rather than a manner best suited for computer translation. The phrase *programming fundamentals* refers to the core ideas of computer programming, including the mathematical foundations, historical perspectives, current and future computing frameworks, the essentials of data structures and algorithms, and the analysis of algorithms using mathematical and experimental methods. This course introduces students to the fundamental concepts of computer programming using a literate programming style.

The course material and grading scheme is designed to suit students with widely varying backgrounds, from novices who use computers only tentatively to experts who happily write device drivers before breakfast. Each student will be assigned reading and writing tasks appropriate to her background.

Two additional points are worth emphasizing: First, although the course is designed to be accessible to a wide range of students, it is a bona fide programming course. Students will use a simple, but powerful and unforgiving, programming language to write computer programs that must run correctly and efficiently. The choice of textbooks reflects this goal. A quote from an *Amazon.com* review of the book by Kernighan and Ritchie is instructive: "Just about every C programmer I respect learned C from this book. Unlike many of the 1,000 page doorstops stuffed with CD-ROMs that have become popular, this volume is concise and powerful (if somewhat dangerous)—like C itself." Similarly, mathematical rigor will be emphasized in the coverage of analysis of algorithms. The accommodation for varying backgrounds will be made by compromising on quantity but not on quality. Second, although the course is programming course, it is one that emphasizes the literary aspects of computer programs. Students will read computer programs written in both the

literate programming and more conventional styles and will examine them carefully from both literary and computing perspectives. Classroom discussions will emphasize proper critical technique, which will be tested using the critiques that are worth a substantial portion of the grade. Similarly, the literate programs written by students will be evaluated not only on their correctness and efficiency, but also on their effectiveness in communicating ideas clearly to a human reader. Such programs are required to go far beyond the routine of commented code (even well-commented code). They must present the program in a compelling and engaging manner with the human reader as the primary concern. The term paper will require students to provide a cogent analysis of one or more topics related to the course.

Textbook and Readings

Required Textbooks:

1. Brian W. Kernighan and Dennis M. Ritchie. *The C Programming Language*. Prentice Hall, second edition, March 1988.

2. Brian W. Kernighan and Rob Pike. *The Unix Programming Environment*. Prentice Hall, March 1984.

Other Readings: This list will be expanded as the course progresses.

1. Jon Bentley and Don Knuth. Programming pearls: Literate programming. *Communications of the ACM*, 29(5):364–369, May 1986.

2. Jon Bentley, Don Knuth, and Doug McIlroy. A literate program. *Communications of the ACM*, 29(6):471–483, June 1986.

3. Derrick Coetzee. An efficient implementation of Blum, Floyd, Pratt, Rivest, and Tarjan's worst-case linear selection algorithm. `http://moonflare.com/`, January 2004.

4. Donald E. Knuth. Mini-indexes for literate programs. *Software—Concepts and Tools*, 15:2–11, 1994.

5. Jon Bentley. Litle languages. *Communications of the ACM*, 29(8):711–721, August 1986.

6. Donald E. Knuth and Silvio Levy. The CWEB system of structured documentation. User manual. `http://www-cs-faculty.stanford.edu/~knuth/cweb.html`, February 2002. Version 3.64.

7. Silvio Levy and Donald E. Knuth. Counting words. Sample program in the CWEB distribution. `http://www-cs-faculty.stanford.edu/~knuth/cweb.html`, August 1995.

8. Donald E. Knuth, Tracy Larrabee, and Paul M. Roberts. *Mathematical Writing*. Mathematical Association of America, Washington, D.C., 1989.

9. Donald E. Knuth and Silvio Levy. *The CWEB System of Structured Documentation*. Addison-Wesley, 1993.

Contact Information

Class meetings:

> **Time:** Tuesdays 3:00–5:30 p.m.
> **Location:** TBA

Instructor: Sudarshan S. Chawathe

> **Office:** Neville Hall, Room 224.
> **Office hours:** (Please check for changes.)
>   TBA
> **Phone:** +1-207-581-3930.
> **Email:** `chaw@cs.umaine.edu`
>   Use email only for messages unsuitable for the newsgroup. (See below.) Please put the string *HON313* near the beginning of the Subject header of your messages to me.
> **Web:** `http://cs.umaine.edu/~chaw/`.

Teaching Assistant: TBA

Online Resources

Class Web site: `http://cs.umaine.edu/~chaw/hon313/`
We will use the class Web site for posting assignments, readings, hints, and other resources. Please monitor it.

Class Newsgroup: We will use the local USENET newsgroup `umaine.hon313` on the NNTP server `news.cs.umaine.edu` for electronic discussions. If you are unfamiliar with USENET, you may find the Web interface at `http://cs.umaine.`

edu/~chaw/news/ useful as a quick way to get started. You may find further information on USENET at http://en.wikipedia.org/wiki/Usenet. The newsgroup is the primary forum for electronic announcements and discussions, so please monitor it regularly, and post messages there as well. Unless there is a reason for not sharing your question or comment, please *use the newsgroup, not email*, for questions and comments related to this course.

Class mailing list: *Please make sure you are on the class mailing list.* A sign-up sheet is circulated at the first class meeting. If you miss it, please contact me to get on the list. We will use this mailing list only for urgent messages because all other messages will go on the class newsgroup. There should be fewer than a dozen messages on this list over the semester.

Policies

Due dates: All due dates (and times) are strict, as announced in class. If you believe your work was delayed by truly exceptional circumstances, let me know as soon as those circumstances are known to you and I will try to make a fair allowance. However, *the default is that you get a zero if you don't turn in the work on time.*

Attendance: Although I expect students to attend all class meetings, I will not be taking attendance. If you miss a class meeting, you are responsible for making up the lost material. If you have a valid reason for missing a class, let me know early and I will try to help you make up the class.

Make-up classes: I may have to reschedule a few classes due to my other professional commitments. I will make every attempt to minimize the number of such occurrences and to reschedule for a time that works for most students. Further, I will make sure no student is penalized by such occurrences.

Special needs: If you have special needs of any kind, including, but not limited to, disabilities, absences due to participation in sports or other activities, etc., please contact me *as soon as the need is known to you* and I will try to accommodate them as much as possible.

Academic honesty: I expect you to hold yourselves to the highest standards of academic honesty. Please take this point very seriously. If you are not sure if something is permitted, check with me. *All help you receive, even if permitted, must be prominently noted in all work you submit.* Erring on the side of giving too much credit is far better than the alternative. Plagiarism and other forms of cheating will result in very stiff penalties (including, but not limited to, an F grade in the course and further disciplinary action from the university).