

BU CAS CS 320 (FALL SEMESTER, 2004)
CONCEPTS OF PROGRAMMING LANGUAGES

Syllabus

- **Semester** Fall, 2004
- **Lecture Times:** TR 9:30-11:00AM
- **Classroom:** KCB 102
- **Instructor:** Hongwei Xi (Office Hours: Thursdays 11-12 and Mondays 3-5)
- **Instructor's Office:** MCS 172, 111 Cummington Street
- **TA:** Alexandra Stefan (TBA)
- **TA's Office:** (TBA)
- **Class Homepage:**
<http://www.cs.bu.edu/~hwxi/academic/courses/CS320/Fall104/classpage.html>
- **Textbook:**
 - *Structure and Interpretation of Computer Programs* (second edition), by Harold Abelson and Gerald Jay Sussman with Julie Sussman. ISBN 0-07-000484-6. McGraw-Hill.
 - Programming in Standard ML (draft), by Robert Harper.
- **Midterm** : A one-hour in-class exam on Thursday, 28 Oct 2004.
- **Final:** A two-hour exam from 12:30PM to 2:30PM, Thursday, 16 Dec 2004.
- **Overview:** Concepts of Programming Languages is a course that introduces students to some fundamental concepts in programming language design and implementation. The primary goal is to allow students who complete this subject to have a good feel for the elements of style and aesthetics of programming and a good command of the major techniques for controlling complexity in a large system.

The programming language Scheme, a dialect of Lisp that we use in this course, is an attempt to bring together the power and elegance of List and Algol. From Lisp we take the metalinguistic power that derives from the simple syntax, the uniform representation of programs as data objects, and the systematic management of heap-allocated data. From Algol we take lexical scoping and block structure.

Standard ML (SML) is a functional programming language that makes extensive use of types in capturing programming invariants. We are to explain how datatypes in SML can be employed to conveniently model data structures and how pattern matching can facilitate programming with datatypes. Also, we are to make use of the module system of SML in the construction of (relatively) large programs.

Ultimately, we would like to make a convincing argument that programming can be a great deal of fun.

- **Grades** The final score is calculated using the following formula.

$$\text{final score} = 40\% \cdot (\text{homework}) + 20\% \cdot (\text{midterm}) + 40\% \cdot (\text{final})$$

The final letter grade is calculated as follows.

- **A:** final score is 85% or above
 - **B:** final score is 75% or above
 - **C:** final score is 65% or above
 - **D:** final score is 60% or above
- **Homework Assignments** There will be a homework assignment every one or two weeks depending on the amount of effort and time needed to finish the assignment. An assignment that is turned in after its due time is accepted but penalized according to the following policy.
 - 10% point deduction if the assignment is turned in within 24 hours after its due time.
 - 25% point deduction if the assignment is turned in between 24 and 48 hours after its due time.
 - 50% point deduction if the assignment is turned in between 48 and 72 hours after its due time.
 - no credit if the assignment is turned in more than 72 hours later after its due time.
 - **Academic Integrity:** We adhere strictly to the standard BU guidelines for academic integrity. For this course, it is perfectly acceptable for you to discuss the general concepts and principles behind an assignment with other students. However, it is not proper, without prior authorization of the instructor, to arrive at collective solutions. In such a case, each student is expected to develop, write up and hand in an individual solution and, in doing so, gain a sufficient understanding of the problem so as to be able to explain it adequately to the instructor. Under *no* circumstances should a student copy, partly or wholly, the completed solution of another student.