16.070 Introductions to Computers and Programming
Spring 2003 Course Syllabus

Introduction to Computers and Programming will provide students with a strong foundation of fundamental programming skills and theory, applications of computers to solve aerospace computational problems, and the use of software in aerospace vehicles and systems. Topics include: digital computer organization, structuring and planning algorithmic solutions to problems, programming and software engineering fundamentals, and introduction to discrete mathematics. Substantial programming assignments using Aero/Astro problems are an integral part of the subject. Progressive homework coding assignments that build upon one another will solidify the software lifecycle process and enable the students to gradually progress from simple workstation programs to a larger software project. A web page will be maintained as the electronic repository of lectures, homework assignments, and class notes. The URL is http://web.mit.edu/16.070/www/

Course Objectives: After attending 16.070 the student will have an understanding of the fundamentals of computer science and will have learnt problem solving using the Ada 95 programming language.

Measurable outcomes (assessment method):

- To be able to solve simple problems in computer science with a specific focus on digital logic, number systems, proof theory, and algorithm analysis (problem sets, quizzes).
- To be able to solve basic programming problems (programming assignments, quizzes).
- To get an intuitive understanding of the process of programming: Problem understanding, formulation, solution, and implementation cycle (programming assignments, final project, quizzes).
- To be able to translate intuitive understanding to practical implementation using good design practices and software tools (Programming assignments, Final Project).
- To develop a programming style that is accepted industry practice (Programming assignments, Final Project).
- To understand the impact of computer science on aerospace (Final Project).

Teaching Staff:

Instructor:	Prof. I. Kristina Lundqvist	Room: 33-332	kristina@mit.edu	x2-2550
Teaching Assistants:	Jayakanth Srinivasan	Room: 33-106	jksrini@mit.edu	x2-3312
	Gerardo Guevara	Room: 33-106	alpha_g1@mit.edu	x2-3312
	Nayden Kambouchev	Room: 33-106	nayden@mit.edu	x2-3312
	Malia Kilpinen	Room: 33-106	malia@mit.edu	x2-3312

Robin Riedel	Room: 33-106	robinr@mit.edu	x2-3312
Catherine Chang	Room: 33-106	cwchang@mit.edu	x2-3312
Mabel Fang	Room: 33-106	mabeline@mit.edu	x2-3312
Chinwe Nyenke	Room: 33-106	chinwe04@mit.edu	x2-3312
Darlene Utter	Room: 33-106	darlene@mit.edu	x2-3312

Time: MWF 2-3, Lecture in 3-270; R 2-3, Recitation in 35-225
One hour Computer Lab Sessions will be held in 33-202 by arrangement, during M 3-5 and T 9-5.

Collaboration: Collaboration, such as working with others to conceptualize a problem or define requirements, is allowed and encouraged as long as it is identified. Plagiarism, such as copying someone else's code, is **not** allowed. See the course website for details.

Grading: Grading for this course will be as follows:

Material	**Percent of Total Grade**
Homework and coding assignments	40%
Final Project	20%
Exams #1, #2, and #3	10%/15%/15% respectively

Textbooks: The following textbooks provide resource material for this course:
1) *Computer Science an overview*, 7th edition, by J. Glenn Brookshear.
2) *Ada 95 Problem Solving and Program Design*, by Feldman and Koffman, 3rd edition.

The course syllabus appears in the following table. There are 14 weeks of class-time covered over 16 calendar weeks (including spring break and finals) that are shown in the first column. The second column describes the material to be covered for each week and a parenthetical note identifying who is the lecturer if it is other than the main instructor. The third column provides the Chapters in Brookshear's (B) or Feldman's books (F). Extra material to read will be distributed later during class. The final column provides a schedule for the Homework assignments. Problem Sets will be posted to the website on Wednesdays and solutions will be due back at the beginning of class the following Wednesday, except when noted otherwise. As a general rule, the lectures introduce general computer and programming concepts, and the recitations further examine these concepts and show examples of how theory can be used in practice.

Wk	Description	Reading	Homework
1			
2/5	W. **Introduction** – The course as part of the Aero/Astro core, Organization of the course, Expectations, Class objectives. What is programming? What is Ada, and why Ada?	B0 F1.7	PS #1 out - 5% of homework grade
2/6	R. Programming mechanics: how to use the PC computers, brief intro to compiler, tools, Source code, executable, libraries, simple I/O, and web page basics (JS)	F1.3, F1.8, F1.9	
2/7	F. **Machine architecture:** Data storage, gates, flip-flops, memory organization, mass storage, representing numbers, text as bit patterns	B1.1-1.4	
2.			
2/10	M. **Ada** – Anatomy of an Ada program, basics of designing a program, software development methods, what is an algorithm, control structures	F1.4-1.5, F2	
2/11	T. LAB		
2/12	W. **Machine architecture:** The binary system, fractions in binary, negative numbers, floating-point notation, generic data compression, Huffman coding, Grey code	B1.5-1.9	PS #1 due 2/12 PS #2 out – 10%
2/13	R. Expected format for homework, steps for programming algorithms (JS)	F1.4	
2/14	F. **Machine architecture:** Computer architecture, machine language, program execution	B2	
3.			
2/17	Presidents Day – Holliday		
2/18	T. (Monday lecture) **Ada:** Intro. to straight-line programs, variables, input/output	F3	
2/19	W. **Ada:** Subtypes, enumeration types, the importance of packages	F4	PS #2 due 2/19 PS #3 out – 10%
2/20	R. Subtypes implementation example (JS)		
2/21	F. **Software:** Single-, multi-processor systems, Operating system architecture, concept of a process, client/server model	B3.1-3.3	
4.			
2/24	M. **Ada:** Decision statements, writing functions and packages, for statement, loops, subtypes (JS)	F5, 6	
2/25	T. LAB		
2/26	W. **Ada:** procedures, exception handling (JS)	F7	PS #3 due 2/26 PS #4 out – 10%
2/27	R. Examples of subprograms, Q&A (JS)		
2/28	F. **Ada:** Scalar data types, the case statement (JS)	F8	
5.			
3/3	M. **Algorithms:** Arrays, ADT, Best, worst, and average cases	Handouts, F9.5-9.10, 11.1, 11.2, B7.1-7.2	
3/4	T. LAB		
3/5	W. **Algorithms:** Asymptotic analysis: upper/lower bounds, Θ notation, Binary search, Insertion sort, merge sort	Handouts	PS #4 due 3/5 PS #5 out – 5%
3/6	R. Simple analysis case studies, Exam #1 review (JS)		

Wk	Description	Reading	Homework
3/7	F. **Data structures:** Stack ADT, queue ADT, access types, linked lists	F15, F16.3-16.6, B7.3-7.5	
6.			
3/10	M. **Exam #1 – in class exam**		
3/11	T. LAB		
3/12	W. **Data structures:** Linked lists, doubly linked lists, circular queues	Handouts	PS #5 due 3/12 PS #6 out – 10%
3/13	R. More on sorting (JS)		
3/14	F. Introduction to graphs and minimum spanning tree algorithms (Prof. E. Modiano)	F15.6, B7.6	
7.			
3/17	M. Information and Entropy (Prof. E. Modiano)	Handouts	
3/18	T. LAB		
3/19	W. **Algorithms:** Recursive functions, recursion	F14	PS #6 due 3/19 PS #7 out – 15%
3/20	R. Graph algorithms, problems on information theory (JS)		Final projects out
3/21	F. **Projects** (IKL, JS)		
8.	SPRING VACATION		
9.			
3/31	M. **Algorithms:** Hashing, hash functions, open hashing, closed hashing	B8.4	
4/1	T. LAB		
4/2	W. **Ada:** Real-time aspects of Ada / Concurrency in Ada (Prof. L. Asplund)	F11	PS #7 due 4/2 PS #8 out – 5%
4/3	R. Hashing functions, Basics of Tasking, Exam #2 review (JS)		
4/4	F. **Ada:** Systems programming using Ada (Prof. L. Asplund)		
10.			
4/7	M. **Data structures/algorithms/Ada**		
4/8	T. LAB		
	Exam #2 – 2 hour evening exam held in TBD		
4/9	W. **Theory of computation:** Finite State Machines/Automata (FSM), definition and examples	B11	PS #8 due 4/9 PS #9 out – 10%
4/10	R. More Concurrency, FSM (JS)		
4/11	F. **Theory of computation:** Non-deterministic finite automata (NFA, DFA), Regular expressions	Handouts	
11.			
4/14	M. **Theory of computation:** Context-free grammar (CFG), push down automata (PDA)	Handouts	
4/15	T. LAB		
4/16	W. **Theory of computation:** The Church-Turing thesis, Turing machines	Handouts	PS #9 due 4/16 PS #10 out – 10%
4/17	R. Ambiguity, Conversion of Regular Expressions (JS)		

Wk	Description	Reading	Homework
4/18	F. **Theory of computation:** Hard problems: NP-Completeness, the halting problem	Handouts	
12.			
4/21	Patriots Day – Vacation		
4/22	Patriots Day – Vacation		
4/23	W. **Proof theory:** Proof by Construction, Contradiction, Induction	Handouts	PS #10 due 4/23 PS #11 out – 10%
4/24	R. Proof examples (JS)		
4/25	F. **Proof theory:** Proof by Construction, Contradiction, Induction (contd.)	Handouts	
13.			
4/28	M. **Software engineering:** The software life cycle, design methodologies, testing	B6	
4/29	T. LAB		
4/30	W. Guest lecture (Prof. N. Leveson?)		PS #11 due 5/1
5/1	R. Exam #3 review (JS)		
5/2	F. Guest lecture (Prof. M. Win?)		
14.			
5/5	M. Guest lecture (Prof. B. Williams?)		
5/6	T. LAB		
5/7	W. **Exam #3 – in class exam**		
5/8	R. Project Q&A (JS)		
5/9	F. TBD		
15.			
5/13	M. TBD		
5/14	T. LAB: Student demo of project		
5/15	W. Future directions, return of Exam #3		
16.	FINAL EXAM WEEK:		