16.070 Introductions to Computers and Programming
Spring 2002 Course Syllabus

Introduction to Computers and Programming will provide students with a strong foundation of the basics of software engineering relevant to Aerospace and aeronautical applications. Students will obtain hands-on experience programming a single board computer to enhance their understanding of embedded systems. 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 an interactive, embedded system. 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: The Course Objectives are set in the context of conceiving, designing, implementing, and operating aerospace/aeronautical information systems. Students will be able to

1) Design modular programs using a top-down design approach (CD);
2) Create structured, well-documented computer programs (I)
3) Test and analyze programs to ensure proper program operation (IO)

Measurable outcomes for this course will be the following:

1) Demonstration of problem solving employing a methodical software development process [measured by problem sets and exams]
2) Conception and design of applications programs (e.g., simulators and real-time embedded systems) [measured by problem sets and exams]
3) Building, testing and operation of real-time embedded application programs through hands-on experience with a single-board computer [measured by problem sets and demonstration]
4) Demonstration of effective test methods to evaluate and document program execution [measured by problem sets and demonstration]

Teaching Staff:

Instructor:	Dr. Lorraine Fesq	Room: 37-435	fesq@mit.edu	(617) 253-8080
Co-instructors:	Prof. Ed Crawley	Room: 33-207	crawley@mit.edu	(617) 253-7510
	Heidi Perry	Draper Laboratory	hperry@draper.com	
	Terry Smith	Draper Laboratory	tsmith@draper.com	
Teaching Assistants:	Louis Breger	Room: 17-010	lbreger@mit.edu	x3-2428
	Kay Sullivan	Room: 17-010	kay@mit.edu	x3-2428
	Gerardo Guevara	Room: 33-106	alpha_g1@mit.edu	x2-3312
	Nayden Kambouchev	Room: 33-106	nayden@MIT.EDU	x2-3312
	Joshua Randall	Room: 33-106	jrandall@MIT.EDU	x2-3312

Graders:	Timothee de Mierry	tim_0248@mit.edu
	Richard Cross	cross@MIT.EDU
	Nick Abercrombie	ncrombie@MIT.EDU
	Robin Reidel	robinr@MIT.EDU

Time: MWF 2-3, Lecture in 3-270; R 2-3, Recitation in 2-190
One hour Computer Lab Sessions will be held in 33-202 by arrangement, during M 3-5 and T 11-5.

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, Parts A/B/C/D	20%
Exams #1, #2, and #3	10%/15%/15% respectively

Textbooks: The following textbooks provide resource material for this course:

1) **Required**: *Engineering Problem Solving with C,* by Etter and Ingber, available at the Coop or on Reserve in the Aeronautics-Astronautics Library.
2) Reference: *Real-Time Systems Design and Analysis: An Engineer's Handbook*, by Phillip A. Laplante, 2nd Edition, available at the Coop or on Reserve in the Aeronautics-Astronautics Library.
3) Reference: *The C Programming Language,* by Brian Kernighan and Dennis Ritchie, on Reserve in the Aeronautics-Astronautics Library. For reference on C syntax.
4) Reference: *Programming Embedded Systems*, by Michael Barr, on Reserve in the Aeronautics-Astronautics Library. How to program embedded systems using C.

The course syllabus appears in the following table. There are 13 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 Etter and Ingber's C book (C) and extra readings that correspond to the material covered. 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 where noted in the outline below. As a general rule, the lectures introduce general computer and programming concepts, and the recitations further examine these concepts and show how they can be implemented in C.

Wk	Description	Reading	Homework
1	Class starts on Wednesday, 2/6/02.		
2/6	W. Introduction - The course as part of the Aero/Astro core, Organization of the course, Expectations, Class Objectives. What is programming? What is C, and why C? *Key section(s) in text:* 1.1	C1, C2	**PS #1 out - 6% of homework grade**
2/7	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		
2/8	F. Crash course in C - Anatomy of a C program, The basics of designing a program, Software development method, what is an algorithm, pseudo-code. 1.3, 2.1		
2			
2/11	M. Variables and Operators - Global/local variables, constants. Operators, expressions, and statements. Data types and ranges. 2.2-2.4, 3.2	C3	
2/12	T. **Lab (33-202):** familiarity with Win2K computers in lab. VC basics. Debugging simple programs. Using variables.		
2/13	W. Program Control - Branch and Jump. *if* and *if else*, relational expressions, *break*, *continue*, and *goto* 3.3		**PS #1 due 2/13** **PS #2 out - 8%**
2/14	R. Explain expected format for homework, i.e., follow software process, include comments and good style**.** Tips for pset 2.		
2/15	F. Program Control - Looping. Why loops? How to terminate loops, conditionals, nested loops. 3.4-3.5		
3			
2/18	M. **Holiday**		
2/19	T. (Monday lecture) Methods for developing algorithms. Alternate ways to represent first approach to solution –flowcharts, state transition diagrams 3.1	C4	
2/20	W. Modular Programming - Functions, function arguments, returning values Basic I/O - character I/O, keyboard I/O, file I/O 2.4, 2.8, 3.6-3.8, 4.1-4.2	C5	**PS 2 due 2/20** **PS 3 out - 10%**
2/21	R. I/O use, algorithm development		
2/22	F. Arrays - defining, initializing, assigning values 5.1-5.3		
4			
2/25	M. Composite Data Types: structures - Defining, initializing, assigning structure values. 7.1-7.2, 7.4 (Draper)	C6, C7	

Wk	Description	Reading	Homework
2/26	T. **Lab:** Implementing Arrays in C.		
2/27	W. Indirect Addressing - Pointer operations, pointers to arrays, pointer arguments - call by reference vs call by value 6.1-6.4 (Draper)		**PS 3 due 2/27** **PS 4 out - 12%**
2/28	R. Implementing pointers in C. Exam Review		
3/1	F. Multidimensional Arrays with pointers and functions 5.8-5.13, 6.2 (Draper)		
5			
3/4	M. **Exam #1 - in class exam.**	Handy	
3/5	T. **Lab:** Hand out Handyboards. Intro to SBC and safe use. How to turn on, how to set up and operate. Warrantee/guarantee agreements	Board Manual	
3/6	W. Developing Software for Embedded Systems - Cross-compilers, development platform, host/target machines, debugging tools		**PS 4 due 3/6** **PS 5 out - 12%**
3/7	R. Steps to use the HB development environment. Interactive C, p-code.	Course Reader:	
3/8	F. Boolean algebra, truth tables, gates	Patt Ch 2, 3	
6			
3/11	M. Data Representation - Bits, bytes, binary number representation, hexadecimal vs decimal. Bit manipulation, error detection and correction (EDAC): SEUs, parity, hamming, SECDED	Course Reader: Patt Ch 4	
3/12	T. **Lab:** Introductory HB exercises		
3/13	W. Number representation - Data types - integer, float, Boolean, character. Float: Accuracy vs precision, overflow, underflow, scaling, bit resolution, quantization. 2.2-2.3, 2.10.		**PS 5 due 3/13**
3/14	R. Arithmetic conversion, ASCII representation, data types and type casting, computer hardware		**PS 6 out - 12%**
3/15	F. Introduction to computer hardware - CPU, memory, I/O. Processor speeds, word size, stacks, registers, etc. 1.2		
7			
3/18	M. Devices and Interfaces - devices that are connected to computers, communication protocols, software needed for communication (Prof. Ed Crawley)		
3/19	T. **Lab:** HB exercises: data types, type casting, bit flicking, Boolean algebra**.**		**PS 6 due 3/20**
3/20	W. Simulation and State Propagation. (Prof. Ed Crawley)		**PS 7 out - 12%**
3/21	R. Steps in developing a simulation program		
3/22	F. Simulation and State Propagation - cont. (Prof. Ed Crawley)		

Wk	Description	Reading	Homework
8	**SPRING BREAK**		
9			
4/1	M. Dynamic Memory Allocation 6.5 (Draper)	C5.4-5.7, C6.5, C7.3	
4/2	T. **Lab:** Dynamic Memory Allocation		
4/3	W. Data structures - stacks, queues, linked lists 7.3 (Draper)		**PS 7 due 4/3**
4/4	R. Implementing Dynamic Memory and data structures		**PS 8 out - 12%**
4/5	F. Handy algorithms - Sorting, searching 5.4-5.7 (Draper)		
10			
4/8	M. Review for Exam		
4/9	T. **Lab:** Practice exam questions		
4/10	W. **Exam #2 - 2-hour evening exam held in TBD**		**PS 8 due 4/12**
4/11	R. Implementing data structures, sorting and searching algorithms		**Final Project out -**
4/12	F. Introduction to the Final Project. Part A: HB controller and Serial I/O; Part B: Integrated controller/simulator; Part C: Integrated controller/vehicle; Part D: Test Report	Final Project Handout	**Part A/B/C/D - 25/30/25/20% of project grade**
11			
4/15	M. **Holiday**	Course Reader: Douglass Ch 2	
4/16	T. **Holiday**		
4/17	W. Introduction to embedded real-time systems. Definition of terms, real-time vs real fast, characteristics of RT systems. (Draper)		
4/18	R. Serial port data transfer		
4/19	F. Real Time computer components and low level features - types of memory, interrupts(Draper)	Course Reader: Memory Types	
12			
4/22	M. Introduction to Real Time Operating Systems. Building blocks of real time OS and executive; polled loop vs interrupts (Draper)	Course Reader: LaPlante Ch. 6	**Part A due 4/22**
4/23	T. **Lab:** Serial I/O between the workstation and Handyboard		
4/24	W. Multi-tasking, foreground/background/full-featured RTOS. Priorities, round robin, pre-emptive scheduling, cyclic executives (Draper)		**PS 9 out - 8%**
4/25	R. Multitasking and priority examples		
4/26	F. Inter-task Communication and Synchronization - buffering data, mailboxes, semaphores, event flags, deadlock. (Draper)	LaPlante Ch. 7	

Wk	Description	Reading	Homework
13			
4/29	M. System Integration. Hardware/software integration, piecewise integration, tips for integrating the final project (Draper)		
4/30	T. **Lab:** Multitasking		
5/1	W. Fault Tolerance - building in redundancy, exception handling, testing (Draper)	Course Reader: L. Alger article	**PS 9 & Part B due 5/1, PS 10 out - 8%**
5/2	R. Question/Answer session on project. Exception handling, testing.		
5/3	F. Optimizing Real-time Software - Handy techniques for increasing code efficiency and speed (Draper)		
14			
5/6	M. Putting it all together - walk through the architecture of a real embedded system (Draper)	Failure Reports	**Demo Part B in Lab**
5/7	T. **Lab:** Student Demonstrations of Final Project Part B.		
5/8	W. Failure reports due to software - NASA MCO, Mars Polar Lander; DOD Milstar launch; Ariane V; Therac 25		**PS 10 & Part C due 5/8**
5/9	R. Review for exam		
5/10	F. **Exam #3 -- in-class exam.**		
15			
5/13	M. Hierarchy of languages. Object Oriented design/programming -- objects, packages (Draper)	C8	**Demo Part C in lab**
5/14	T. **Held in Bldg. 33 Hangar**. Student Demonstrations of Final Project Part C.		
5/15	W. Future directions		**Part D due 5/15**
16	FINALS WEEK: **NO FINAL EXAM**		