Welcome to 16.070 - Introduction to Computers and Programming for Aero/Astro Majors

• Introductions

• 16.070 as part of the Aero/Astro core curriculum

• Course Objectives

• Administrative Issue - Computer Lab Sessions to be scheduled

• An overview of the role of software in aerospace applications
Course 16.070 vs Other Computer Courses

- **Aero/Astro**
  - **Numerical Analysis / Numerical Methods / Scientific Computing**
    - 1.00, 10.001, 10.002J, 13.002J, 16.901, 18.330
  - 16.070
  - MATLAB, MAPLE, XESS, 16.900
  - **Hardware**
  - **Operating Systems/Compilers**
  - RMA
  - **Theory of Computation**
    - 6.044J, 6.045J
  - 6.034
  - **Artificial Intelligence**
  - **Algorithms**
    - 6.046J
  - **Languages**
    - 6.001, 6.821
  - **Assembly, Jovial, Ada, C**
  - **Tools**
  - **LISP**
  - **Real-time Embedded Systems**
  - 6.033, 6.035
  - 6.004, 6.832
  - **Numerical Analysis / Numerical Methods / Scientific Computing**
  - 6.001, 6.821
Course Objectives

• Course objectives set in context of conceive, design, implement and operate aero/astro information systems. Students will be able to
  ➢ Design modular programs using a top-down design approach (CD);
  ➢ Create structured, well-documented computer programs (I)
  ➢ Test and analyze programs to ensure proper program operation (IO)

• Measurable outcomes for this course
  ➢ Demonstration of problem solving employing a methodical software development process [measured by problem sets and exams]
  ➢ Conception and design of applications programs (e.g., simulators and real-time embedded systems) [measured by problem sets and exams]
  ➢ 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]
  ➢ Demonstration of effective test methods to evaluate and document program execution [measured by problem sets and demonstration]
Expectations

- Class/Recitation attendance not mandatory, but highly recommended
- Homework is 55% of your grade -- do it, and hand it in on time!
  - Late assignments accepted only if you contact TA Thomas Jones before assignment is due, and discuss circumstances and plan for completion
  - Late assignments will be docked 1/3 of grade for each 24 hours late, and will not be accepted after solutions have been posted
- Collaboration/teamwork policy on the web
- Pathway for feedback
  - "Muddiest Point" and "Comments" on the web, both anonymous
  - Concerns - Dr. Fesq, TA (grad or undergrad), advisor, Prof. John Deyst
- Reading assignments - read ahead to understand lectures and to accomplish homeworks/exams.
Administrative Issue

• Scheduling "Tuesday" Computer Lab Sessions
  ➢ ~85 students, 6 sessions available, ~15 students per session
  ➢ Lab times
    – M3, M4
    – T9, T10, T11, T12
  ➢ Go to web page and identify conflicts that you have with the lab times
    – Announcements link
  ➢ Lab times will be assigned on Friday, 2/9/01
Overview of Software in Aeronautics/Astronautics

• Software is an integral part of aerospace systems, controlling virtually every movement
  ➢ Space Shuttle has 5 on-board processors, >1 million lines of code
  ➢ F-16D (late '80s) has 15 computer systems, 300 digital processors, 236,000 lines of code
  ➢ B-2 has >200 processors, ~5 million lines of code
  ➢ B777 has ~1400 processors, ~5 million lines of code

• Ground-based systems are even larger
  ➢ FAA Air Traffic Control System to replace 1972 system
    – Begun in 1982, still not completely installed
    – >$6B spent
  ➢ Iridium - 17M lines of code
Software Development in Aeronautics/Astronautics

• In this course, you will learn the principles of software engineering as they relate to two main areas relevant to aerospace/aeronautics
  ➢ Simulations
    – Simulating/modeling real-world applications such as spacecraft/aircraft vehicle dynamics
  ➢ Embedded Flight Software
    – Writing software that control spacecraft/aircraft and ensuring robustness and timeliness
Aerospace Software Engineering

• Top 10 technical issues of concern for aerospace applications software
  ➢ It is large
  ➢ Solves/models complex mathematical/physical problems
  ➢ Uses every available machine cycle and every storage bit when completed
  ➢ Tailored to a single mission
  ➢ Embedded within complex system of many interacting "hardware" elements
  ➢ Embodies significantly new functions never before coded in software
  ➢ Has a stressful operating environment
  ➢ Has complex interactions with highly trained users
  ➢ Its builders do not use it; its users do not build it
  ➢ Its operation risks human life and/or great economic loss

• Aerospace poses many unique and demanding challenges to software
Embedded Systems

• An **Embedded System** is a software system that is completely encapsulated by the hardware that it controls.

• General Purpose Computers have the following properties
  ➢ Stand-alone system, responds primarily to a user
  ➢ Comprised of additional mechanical components such as disk drives
  ➢ **Not** designed for a specific function -- designed for multi-purpose
    – E.g., Spreadsheet, Presentation material, Programming

• In contrast, Embedded Systems have the following properties
  ➢ Exist inside a larger system with purpose of helping system accomplish responsibilities. May interact with users
  ➢ Are connected to monitoring and/or controlling hardware
    – E.g., Temperature sensors, Ailerons
  ➢ Designed to perform a specific function, e.g., microwave oven, ATM
Embedded Systems - cont.

- An Embedded System frequently is a component within a larger system
  - E.g., Modern Automobile
    - An embedded system controls anti-lock brakes
    - An embedded system monitors and controls vehicle's emissions
    - An embedded system displays information on the dashboard
Real-Time Systems

• A Real-Time system is defined as one in which timeliness is as important as the correctness of the output

• Real-Time Systems are often embedded systems

• Real-Time Systems have timing constraints
  ➢ Partly specified in terms of ability to make certain calculations or decisions within a pre-determined time-frame
  ➢ Pre-determined time-frame is called a deadline
    – "When a command is sent to a thruster, the thruster must respond within 10 ms +/- 2 ms"
  ➢ A missed deadline is just as bad as a wrong answer
    – A real-time system as part of an airplane's flight control system

• Designers of Real-Time Systems must guarantee reliable operation of software and hardware under all possible conditions
Programming Languages for Embedded Systems

- A limited number of programming languages are used for embedded systems
  - C - closest to a "standard" for embedded systems
  - Ada - developed by the Department of Defense, widely used in military applications
  - Java - becoming more popular, but not proven for real-time systems
  - Assembly - lower-level language unique to the processor being used (i.e., not portable)

- In this course, we will use C for our programming development
  - Most broadly applicable language when considering embedded and non-embedded applications
  - Easy transition to C++
  - Can be used to program the Handyboards
Summary

• What you should have learned today
  ➢ Why software is important in aerospace
  ➢ What makes aerospace uniquely challenging to developing software
  ➢ An embedded system is ________________________________
  ➢ A real-time system is _________________________________
  ➢ Review reading assignment C1 and R1 with today's lecture

• For Thursday
  ➢ Fill out web-based form to sign up for lab session
  ➢ Check that your computer account is working in both NT clusters -- buildings 33 and 37

• Friday
  ➢ Crash course in C -- Read C3 in preparation for lecture