

## Course Overview

Policies, Grading & Outline

CIT 593  
Fall 2007

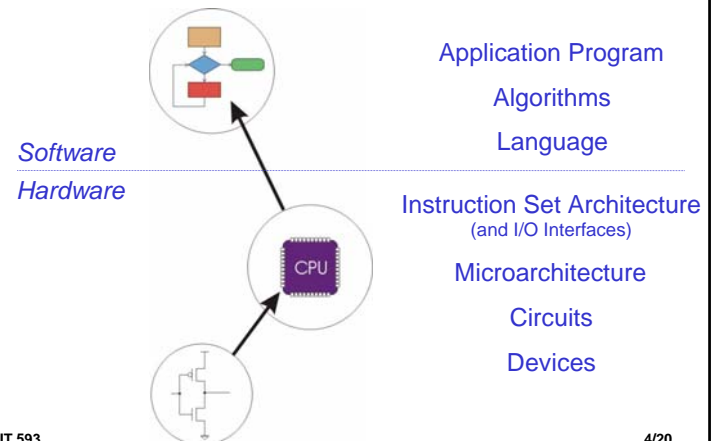
## About CIT 593-95 in General

- ✓ How computers work
- ✓ What computers consist of
- ✓ How they are organized internally
- ✓ What are the design tradeoffs
- ✓ How design affects programming and applications
- ✗ How to fix computers
- ✗ How to build myself one real cheap

## Why take CIT 593-95 sequence?

- Understand how computer represent and process information
- Write machine dependent software
  - E.g. compilers, operating systems, and device drivers
- Interested in high performance studies
  - E.g. how to arrange/design programs to gain maximum performance from a given architecture
- Benchmarking
  - E.g. compare computer architectures for a purchase decision as a member of the IT computer purchasing team.

## Computer Organization



## Overview of Topics CIT 593-95

- **Hardware**
  - Data Representation, von Neumann machine model , transistors, gates, digital logic
- **Instruction Set Architecture (ISA)**
  - Basic Operations that machine can perform
- **Assembly language**
  - Structured programming which is specific to a machine
- **C programming**
  - Language features + relationship to assembly language
- **Operating System**
  - A management program that allows application programs to interface the machine
- **Input/Output**
  - How does the computer communicate with the outside world?
  - How does simultaneous interaction with the computer work?

## Why learn Hardware ?

- **To learn the foundations**
  - Problem stated in programming language is actually solved by electrons moving around inside the electronics of the computer!!
- **To know its limitations**
  - What makes a computer system tick before you can attempt to optimize programs ?

## Why learn Assembly ?

- Computers can only understand voltages and abstractly we represent them in binary notation (0 or 1)
  - We call this machine language
  - Gets very tedious to program
- Assembly is the lowest form of human-readable notation for the machine language
  - Better able to understand & appreciate a high-level language



## Why Learn C Language ?

- **What is C?**
  - High-level language
  - In between assembly and Java/VB/C#
- **Very common**
  - Operating systems and even general applications
  - Foundation for C++/C#/Java
  - Still widely used in conjunction with C++ by many companies
  - Assembly-to-C migration for embedded applications

## Why Learn about OS & I/O

- You use these in day to day when you interact with the computer
- Get perspective on what is happening
  - E.g.1 When interacting with multiple programs
  - E.g.2 Blue Screen of Death



```
A problem has been detected and Windows has been shut down to prevent damage to your computer.
The problem seems to be caused by the following file: smoboom.sys
smoboom.sys is a new installation and your hardware or software manufacturer may have released updates you might need.
Check to make sure any new hardware or software is properly installed. If this is a new installation, you may want to disable or uninstall the hardware or software. Disable or remove any newly installed hardware or software. Disable any third party memory options such as caching or shadowing. If you need to use Safe Mode to remove or disable components, restart your computer at least five times to select Advanced Startup options, and then try again.
Additional information:
*** STOP: 0x0000007B (0x0000007B, 0x00000000, 0x00000000, 0x00000000)
*** smoboom.sys -- address F8E70217 base at F8E70000, date stamp 3-06-08P7.
```

## Course Topics for CIT 593

- **Bits and Bytes**
  - How do we represent information using electrical signals?
- **Instruction Set Architecture**
  - Formal specification of all the functions a particular machine can carry out
- **Assembly Language Programming**
  - How do we use processor instructions to implement algorithms?
  - How do we implement high-level programming constructs?
    - Modular, reusable code (subroutines), OOP
- **I/O, Traps, and Interrupts**
  - How does processor communicate with outside world?
- **C Programming**
  - How do we write programs in C?

## Transition to CIT 595

- **Implementation**
  - How are instructions actually carried out using transistors?
  - Are there more than one way of building a computer?
    - Micro-architecture
- **Memory Hierarchy**
  - Different kinds of computer memory
- **Overview on the role of the Operating System**
- **Some Advanced Topics in Computer Architecture**

## Staff

- **Instructor: Diana Palsetia**
  - Email: palsetia@cis.upenn.edu
  - Location: Moore 174
  - Office hours: TBA
  - course web:  
<http://www.cis.upenn.edu/~palsetia/cit593f07>
  - Blackboard: <https://courseweb.library.upenn.edu/>
- **TA: Fatima Boujarwah**
  - Email: faa@seas.upenn.edu
  - Office location and hours: TBA

## Textbooks

### ■ Required

- Introduction to Computing Systems from bits & gates to C & beyond
  - Author: Yale N. Patt and Sanjay J. Patel
  - ISBN: 0072467509 (2nd Ed)



### ■ Recommended

- The C Programming language (2nd Edition, Paperback) by Brian W. Kernighan, Dennis Ritchie

### ■ Online Resources

- Plenty of online resources
- See course website

<http://www.cis.upenn.edu/~palsetia/technicalFAQ.html>

## Grading

- Course grades will be curved
  - 90% or above is not necessarily an A
- Assignments (45% of grade)
  - Written and Programming Assignments
- Quizzes (10 % of grade)
  - This is to ensure you keep pace with the class
  - Cover assigned readings not taught in class
- Two exams (45% of grade)
  - Midterm (20%) & Final (25%)
  - Exams will be closed book

## Dates

- Midterm exam: 23rd Oct (during class time)
- Final exam: According to final week schedule
- Assignments (regularly check website)
  - Programming/Homework: As they are posted
  - Quizzes will be announced a week in advanced

## Assignment Grading

- If submitted later than due date
  - 10% per day for five days and then no credit
  - Weekend days count as late days
- Program grading is generally
  - Good documentation 20%
  - Sound structure 20%
  - Correct output or results 60%
- Turning assignments
  - Submitted via Digital Dropbox via Blackboard
  - Instructions on how to submit are provided in the syllabus (see course website)

## Academic Honesty

- Appropriate
  - Discuss the assignments with one another to clear doubts
  - Help others debug their work
- Inappropriate
  - Work together unless stated otherwise
  - Copy another's code, or allow your code to be copied
  - Lend your code to someone else, or allow them to copy it
  - Use any code from a book or the web without my permission
- Penalty for first offence:
  - You will be reported to the Office of Student Conduct
- If you may have accidentally broken a rule, talk to Instructor or TA immediately

## How much time to put into the course ?

- On average, at least **10 hours** per week, assuming:
  - You take 3-4 courses
  - You are a full-time student
  - Your studies occupy a 40-hour work week
- 3.5 hours in class + office hours
- 3.5 hours on projects and practice
- 3 hours on readings

## Computing Facilities at Penn

- Penn Eng.:
  - <http://www.seas.upenn.edu/cets/answers/>
  - Information on accounts, labs, software
- If working from home
  - Mac Software: Fetch, DataComet-Secure X
  - Windows: Filezella, SecureCRT

## Questions?

