What’s CSE 240 All About?!

This is not your father’s CSE 240
• Not at 9am!!!
• New approach (bottom up)
• New textbook
• New instructor
• Well integrated into curriculum
• Cooler assignments (demo coming later)
• No assumption that you know C programming
• Second time around

Introduction to computer architecture
• How is data represented?
• What are the pieces of a computer?
• How do computers work?

Programming
• How do I “talk” directly to the machine?
• How do I program in C?

Computer Systems and Computation
• How do simple HW/SW elements come together to realize complex computations?
Big Picture

Hardware
- Representing data, transistors, gates, digital logic structures, von Neumann machine model

Assembly language
- Instructions, (structured) programming, input/output, relationship to hardware

C programming
- Syntax, operators, control structures, functions, pointers, recursion, data structures, relationship to assembly language

Assembly Language Programming

What is an instruction?
- Basic unit of (SW) computation
- Very primitive
- E.g., in LC-3: addition, logical-and, logical-not, branch, load-from-memory, store-to-memory

Focus: Where hardware meets software
- We will examine the hardware that executes instructions
- We will compose instructions to create software

Really little example
```
LOOP JSR TIMER_TICK
ADD R1,R1,#-1
BRp LOOP
```

Today

Objectives

Summarize course implementation
- Background/Prerequisites
- Lectures/Reading/Quizzes
- Homework
- Exams
- Grades

Demo
- Sample homework/project!

Why Take CSE240?

Foundational
- Intersects all aspects of computing

Preparatory/Complementary
- CSE 371: Digital Systems Organization and Design
- CSE 380: Operating Systems
- CSE 341: Compilers and Interpreters
- CSE 260: Mathematical Foundations of CS

Fun!!!
- Who wouldn’t want to know the magician’s tricks?
Objectives
Understand role & relationship of hardware and software

Exposure to . . .
- Machine organization (CSE 371 prep)
- Assembly language programming (CSE 341 prep)
- C programming (CSE 380 prep)

Able to actually build entire (slow) computing system
- Hardware and software
- You’ll get a chance in CSE 371/380

Be distinguished from mere programmers

Why Study Hardware?
Important
- Floaters can’t build effective systems!
- Still drives industry

Timely
- IA-64, SSE, hyper-threading, security, . . .

Opens doors
- Yet another option!

Impress E!

Why Learn Assembly Programming?
Helps understand capabilities of machine

Can be used to exploit processor-specific extensions
- E.g., Pentium with MMX or SSE

Many system components written in assembly
- E.g., microcontrollers, device drivers, DSP code

It’s in the news!
- E.g., stack smashing

Why Learn C Programming?
What is C?
- High-level language (than assembly anyway)
- “Portable assembly language”
- In between assembly and Java/VB/C#

Very common
- Operating systems and even general applications
- Foundation for C++/C#/Java
- Assembly-to-C migration for embedded applications
CSE 240 Implementation

Background/Prerequisites
Lectures/Reading/Quizzes
Exams
Homework
Academic Integrity
Grades

Background/Prerequisites

Requirement: Strong background in programming
- CSE 120
- CSE 110 (A- or better strongly recommended)
- 5 on AP CS exam
- Pass U Penn CIS placement exam

Why?
- Fast pace
- Assume you can program/debug

Lecture

Expectation
- Read appropriate sections in textbook before lecture
- See class schedule for reading assignments

Quizzes
- Complete easy online quiz before each class
- Can work ahead (do a week at once)
- Experimental (appears to be effective)

Lectures
- Will not simply “cover” the material
- Will focus on the “hard stuff”
- Will not stand alone, instead build on reading
- Will be interactive

Exams

Midterm
- Covers digital logic and some assembly (approx. 1st third)
- Tentatively scheduled: Friday 21 October in class
- Open book

Final
- Covers assembly and C (last 2 thirds) and digital logic
- Tentatively scheduled: Wednesday 14 December
- Open book
Homework
Paper and pencil assignments
• Problem solving
• Great exam preparation

Programming projects
• Simple exercises
• Challenging projects (Breakout!)

Discussion
• Encouraged! (TAs, discussion group, etc.)
• Work must be completed alone
• Important: Gilligan’s Island Rule
• Okay: discuss meaning of problem, discuss approaches
• Not Okay: comparing answers, solving questions together

Academic Integrity
The rule is simple
• Claiming another’s work as your own will ruin your life
• See syllabus for details and examples

Who will know?
• We will (inspection, similarity detectors, exams)
• Your friends will
• Your parents will
• You will

Analogies
• Cheating is like going 150 MPH over speed limit while drunk
• Similar consequences (legal and educational)

Remember
• If you need to cheat now, you’ve got much bigger problems

Grades
Final: 30%
Midterm: 25%
Homework: 35%
• Three extensions allowed (see syllabus)
Quizzes: 5%
Participation, attendance, etc.: 5%

Demo
Homeworks 6 - 7
• Build Breakout in assembly language!
Schedule

See web page
• www.seas.upenn.edu/~cse240
• Subject to change

Part I: Hardware (Digital Logic)
• Paper and pencil assignments

Part II: Assembly Programming
• Substantial programming project

Part III: C Programming
• Assume already familiar with HLL programming (but not C)

Next Time

Lecture
• Chapter 1: Introduction to computer systems
• Chapter 2 - 2.2: Integer data types

Reading
• Chapter 1

Quiz
• As always, online; due before start of class

Upcoming
• Homework 1 due Friday 16 September