CMPSC 311 - Introduction to Systems Programming

Introduction

Professors:

Sencun Zhu and Suman Saha

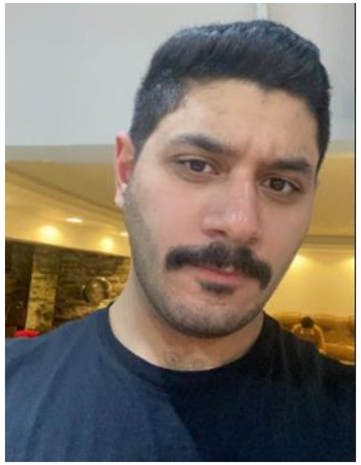
(Slides are mostly by *Professor Patrick McDaniel* and *Professor Abutalib Aghayev*)

Course Staff

Professor:
Sencun Zhu

Professor:
Suman Saha

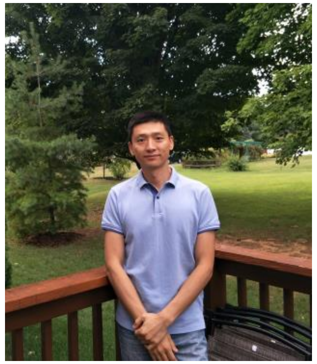
TA: Neeraj Karamchandani


TA: Ali Noureldeen

TA: Shai Sundar

TA: Minli Liao

TA: Thrupti Raj


TA: Ma Qian

TA: Lay Patel

TA: Shakya Chakrabarti

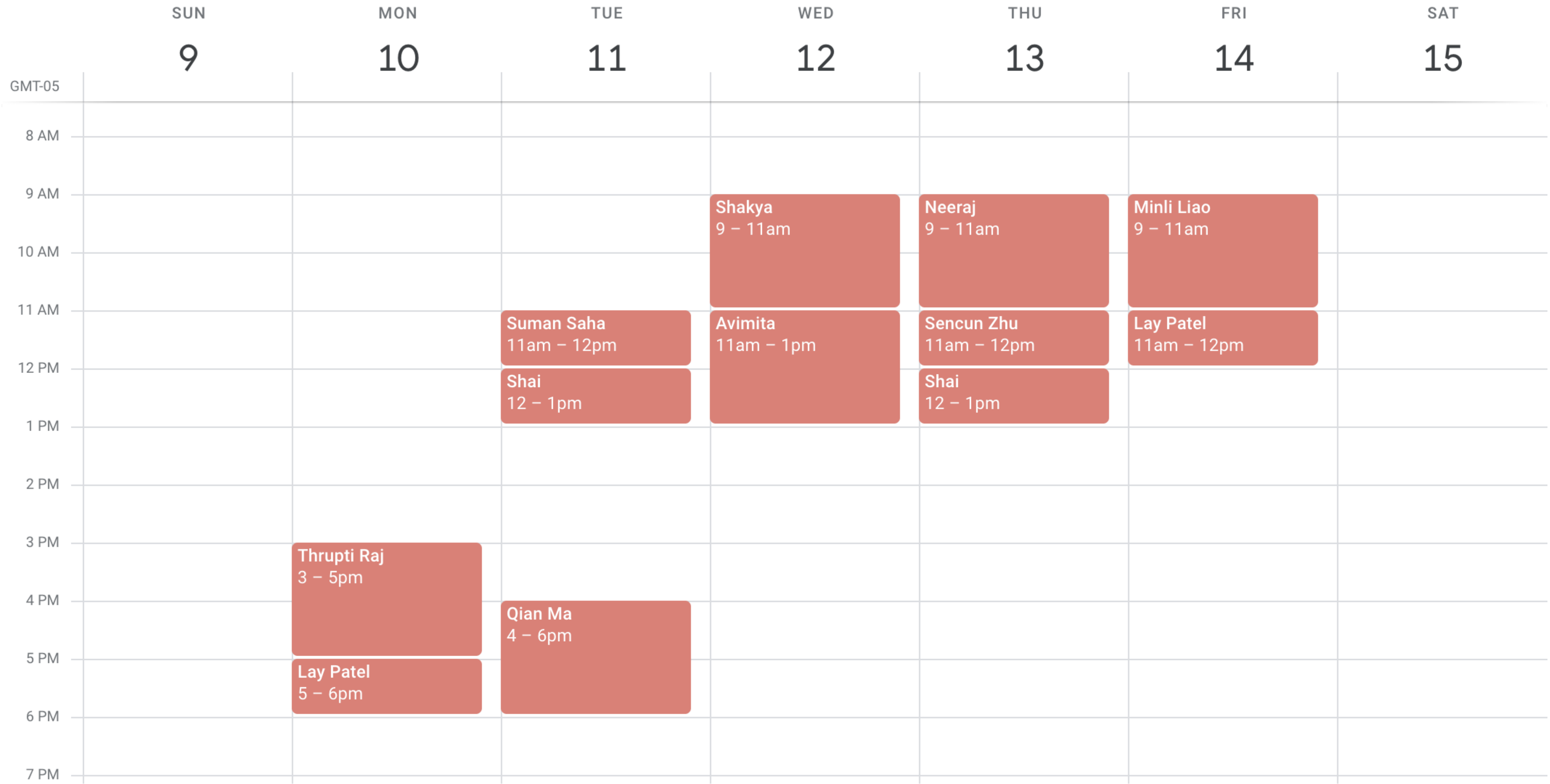
TA: Avimita Chatterjee

Grader: Nishant Asati

Logistics

- Meetings: Monday, Wednesday, Friday: 1:25-2:15PM (Section 1) or 2:30-3:20PM (Section 2), 3:35-4:25PM (Section 3)
- Attendance is not mandatory, but recommended
- Textbook: None
- Communication:
  - Piazza
    - signup: piazza.com/psu/spring2022/cmpsc311
    - everyone can post, ask questions about lectures/assignments
    - Be nice, and you need to search past threads before posting
    - Do not share your code or your program output and it is okay to answer your fellow students' questions
  - Canvas
    - First send emails to your individual TA for any questions.
    - Contact the instructors outside their office hours only when your TA cannot answer your questions (remember we have 450 students!!!! Instructors will only answer emails that cannot be answered by TAs. In this case, you need to copy your TA in the email)

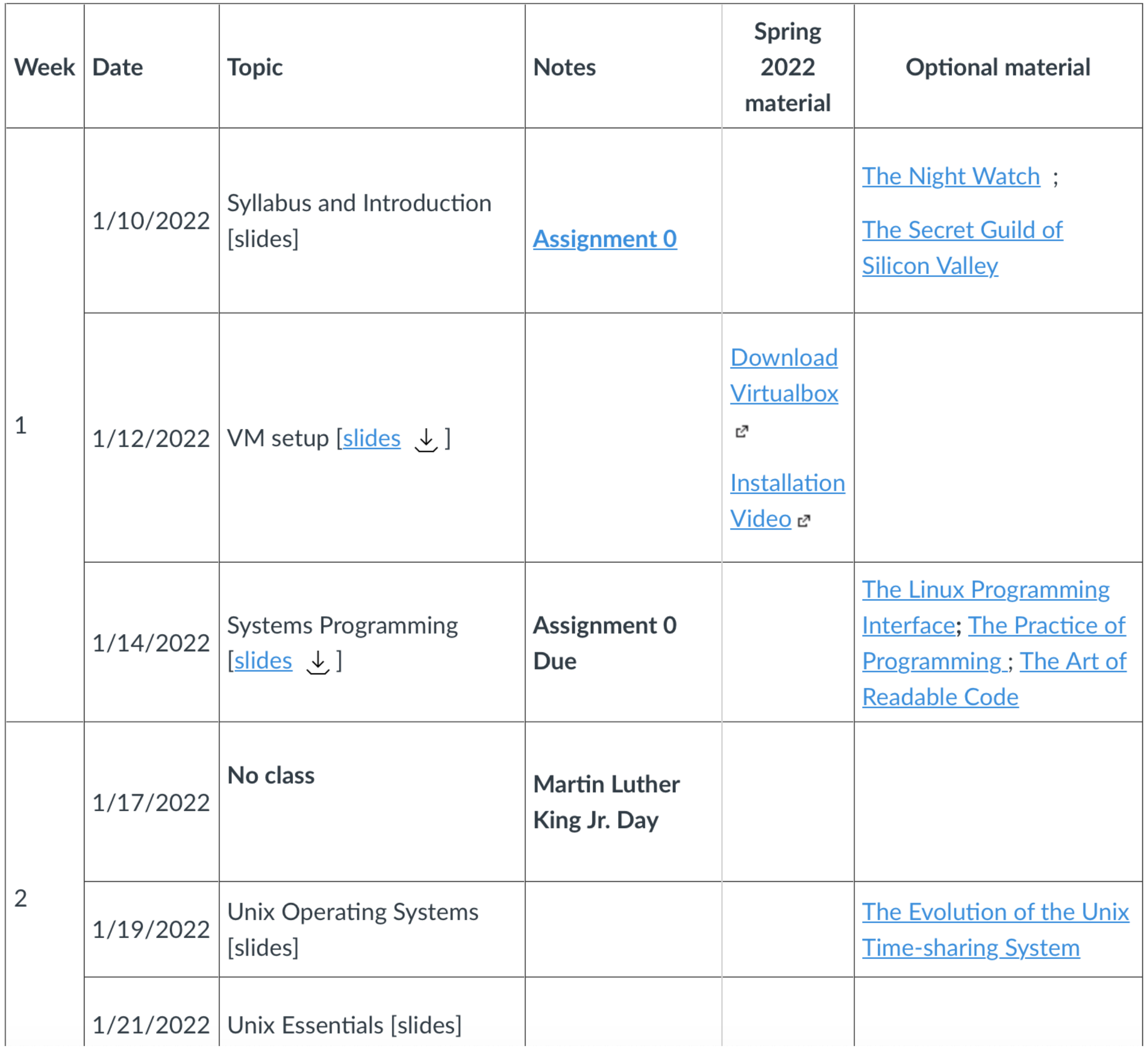
Office hours (Not Finalized Yet)



GMT-05	SUN 9	MON 10	TUE 11	WED 12	THU 13	FRI 14	SAT 15
8 AM							
9 AM				Shakya 9 – 11am	Neeraj 9 – 11am	Minli Liao 9 – 11am	
10 AM							
11 AM			Suman Saha 11am – 12pm	Avimita 11am – 1pm	Sencun Zhu 11am – 12pm	Lay Patel 11am – 12pm	
12 PM			Shai 12 – 1pm		Shai 12 – 1pm		
1 PM							
2 PM							
3 PM		Thrupti Raj 3 – 5pm					
4 PM			Qian Ma 4 – 6pm				
5 PM		Lay Patel 5 – 6pm					
6 PM							
7 PM							

Course Schedule

- Linked from course home page
- Provides links to all of material
  - Slides
  - Assignments
  - Zoom recordings
  - etc.
- Tentative
  - Changes (if any) will be announced



Week	Date	Topic	Notes	Spring 2022 material	Optional material
1	1/10/2022	Syllabus and Introduction [slides]	**Assignment 0**		The Night Watch ; The Secret Guild of Silicon Valley
	1/12/2022	VM setup [slides ↓]		Download Virtualbox ↗ Installation Video ↗	
	1/14/2022	Systems Programming [slides ↓]	**Assignment 0 Due**		The Linux Programming Interface; The Practice of Programming ; The Art of Readable Code
2	1/17/2022	**No class**	**Martin Luther King Jr. Day**		
	1/19/2022	Unix Operating Systems [slides]			The Evolution of the Unix Time-sharing System
	1/21/2022	Unix Essentials [slides]			

Format of Meetings

- Students with conditions that make it difficult to wear a mask (Case 1) or need quarantine due to your own infection or your exposure to infected people or other types of sicknesses (Case 2) may participate in class remotely via Zoom.
  - For Case 1, you need to get approval from the instructors in the beginning of the semester.
  - For Case 2, you may get a period of permissions for Zoom meetings when requested.
  - Per university policy, we cannot open up Zoom meeting as a substitute for in-person meeting, so if your situation does not fall into either of the two cases, you will not be admitted into the Zoom room from the waiting room.
- We will only record one of the three sections via Zoom throughout the semester so that students may watch the missed lectures or review the content.
- The three exams are to be taken in classroom only. No online option will be offered.

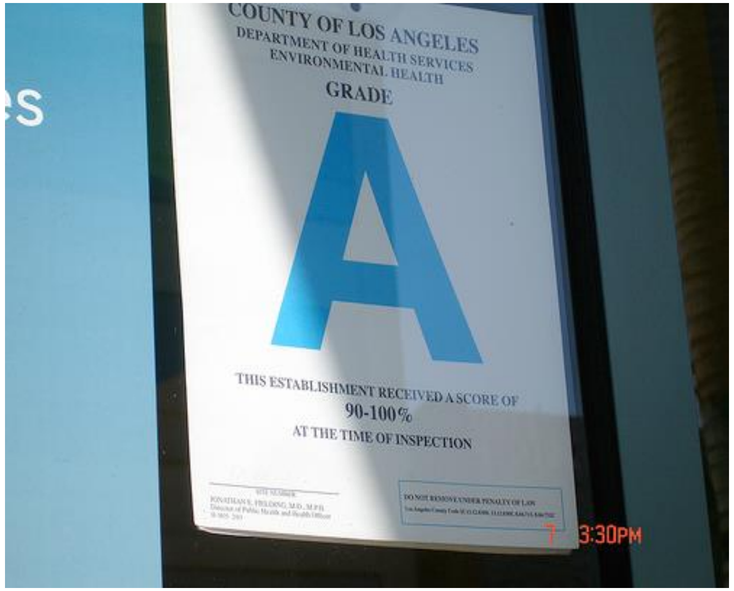
Mask Wearing Policy

- Everyone -- including the instructor and TAs -- are required to wear a face mask in University buildings, including classrooms and labs. You MUST wear a mask appropriately (i.e., covering both your mouth and nose) in the building if you are attending class in person.
- You are not permitted to consume food or drink in classrooms, except for water. If you must drink water, please be especially conscious of maintaining social distancing and minimizing the time your mask is moved aside. Or, better yet, use a straw.
- Anyone attending class in person without a mask will be asked to put one on or leave.

Grading

- The course will be graded on exams, assignments, projects, and class participation in the following proportions:

Percentage	Activity
50%	Course Projects/Exercises
15%	Mid-term Exam
15%	Mid-term Exam
20%	Final Exam (time/date TBD)



COUNTY OF LOS ANGELES
DEPARTMENT OF HEALTH SERVICES
ENVIRONMENTAL HEALTH
GRADE

A

THIS ESTABLISHMENT RECEIVED A SCORE OF
90-100%
AT THE TIME OF INSPECTION

NOTE:

1. You must receive at least 40% of the points for the exams to pass the class.
2. You must receive at least 40% of the points for the class projects to pass the class.

Tools

- Virtual Machine and Guest OS
  - Virtualbox 6.1.30: https://www.virtualbox.org/
  - Ubuntu 20.04.3: https://ubuntu.com/download/desktop
  - Installation video demo: https://youtu.be/x5MhydijWmc
- Git & GitHub
  - We will use git & GitHub
  - "My computer crashed", "Cosmic rays flipped bits of my SSD", etc. – invalid excuses
- Emacs
  - I will cover emacs but it is not required – you can use VScode or other editors
- Many other tools, including bash (shell scripting), make (build automation)
  - I will try to cover a tool before its use is required by the assignment

Course Labs (Projects)

- 6 course labs(projects) planned
  - About 2 weeks to complete each project
  - You individually develop some non-trivial systems functionality
  - Progra
  - May ne
- Progran
  - Correc
  - Clarity/
  - Docum
  - Other f

Policy: All course projects are to be carried out individually. Students are explicitly not allowed to share information, source code, or even discuss the contents of the projects. Students are also forbidden from copying code from the Internet. *Any violation* of these policies will be considered a violation of ethical conduct (cheating) and will result in a zero for the assignment and a full letter grade off the final grade. There will be *no exceptions* given for any reason.

- Lateness policy
  - 10% per-day late penalty, up to 3 days. No exceptions. Period.

MOSS

- Moss (for a Measure Of Software Similarity) is an automatic system for determining the similarity of programs. To date, the main application of Moss has been in detecting plagiarism in programming classes. Since its development in 1994, Moss has been very effective in this role. The algorithm behind moss is a significant improvement over other cheating detection algorithms (at least, over those known to CS folks).

Try it: http://theory.stanford.edu/~aiken/moss/

Assignment #0 (ungraded)

- Go to the course website and find assignment #0 and do what is says. This will only take about 5 minutes or so and does not require technical effort.
- You must complete the assignment by the 14th of January. Failure to meet this deadline will mean you will be dropped from the class.
- I highly recommend you do this immediately!

What I expect you to know today …

- Basic to intermediate programming
  - Java (CMPSC 221)
  - Simple logic and algorithms
- Data structures
  - data organization, management, and storage format that enables efficient access and modification
  - Such as byte, array, list, record (hash table, binary tree) etc.
- Basic computer science literacy
  - CPU, disk drive, file systems (files, folders/directories)

Systems programmer…

- The Night Watch: systems programmer
  - has traced a network problem across 8 machines 3 time zones…
  - has read the kernel source to better understand the universe…
  - has submitted a kernel patch to restore balance to The Force…
- Systems programmer produces software on top of operating systems and hardware
  - Languages: C, C++ (recently Rust, D, and Go)
  - Deeply understands operating systems and hardware
  - Knows *how things work* and how to fix them when they break
- Why not use a modern language with garbage collection?
  - Here’s the answer: Pointers are real. They are what hardware understands.
- If you find yourself drinking a martini and writing programs in garbage-collected, object-oriented Esperanto…
  - it is because there are systems programmers who made it possible…

Class Structure

Systems Programming Basics

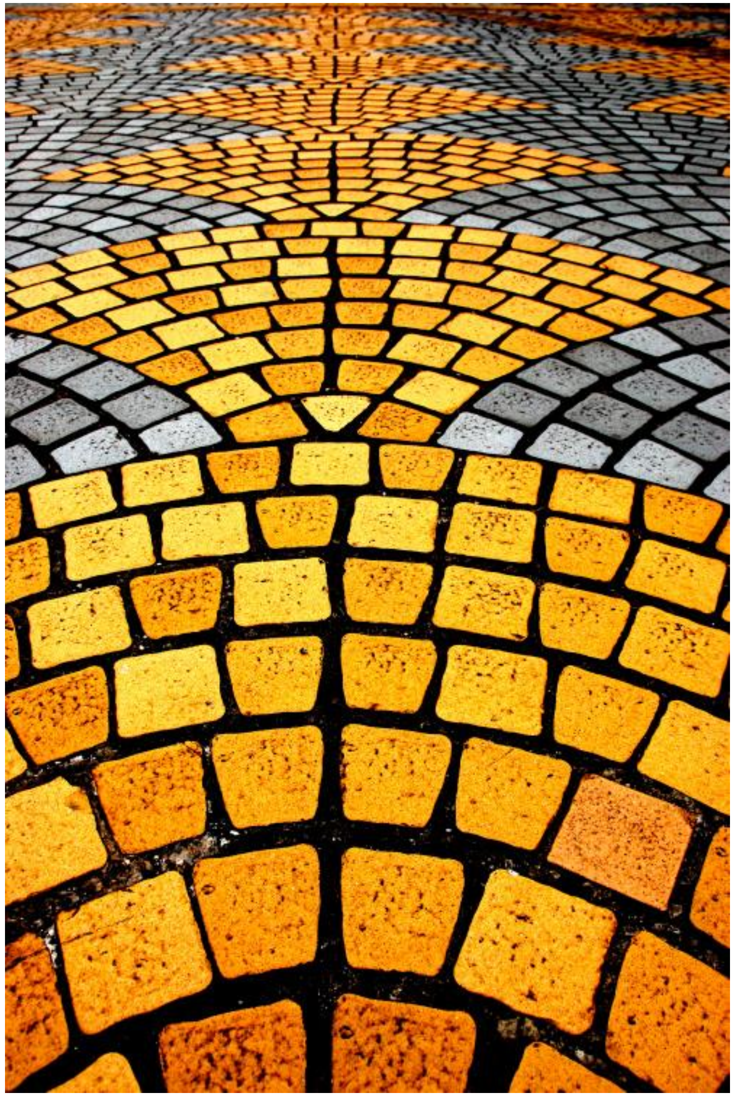
- Systems architecture
- Systems programming (C)
- Systems administration
- Version control systems

Systems Programming Tasks

- Debugging
- Profiling

Advanced Systems Programming

- Concurrency
- Network programming
- Third-party library integration



Goals

- My goal: *to provide you with the tools to professionally develop systems*.
  - Basic technologies
  - Engineering/performance/feature trade-offs
  - How to design, develop and maintain code
- *This is going to be a hard course*. The key to success is sustained effort. Failure to keep up with materials and project will likely result in poor grades, and ultimately little understanding of the course material.

- Pay-off: systems programming skills are essential to a career in computer science related professions.

The Secret Guild of Silicon Valley

- Ignoring the stereotypes and other issues, the article has a point:
  - Expert systems programmers are indispensable
  - You cannot scale a tech company without them
  - Expert systems programmers are always in high demand

- This course is the first step
  - becoming an expert takes time and effort