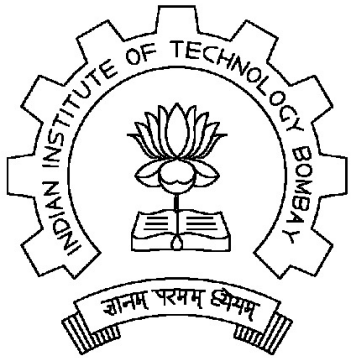
CS699
Software Foundation Lab
Introduction

Kavi Arya
CSE Dept., IIT Bombay



Plan

- Overview:
  - Technology landscape of SW from CSE PG POV
- Teaches you to:
  - Script
  - Build web page
  - Document your work (theory and code)
  - Develop system programming skills
  - Put it all together in a project
- All to help you with later courses/career

Outline

- Course content & goal
- Instructor's take
- Course Notes/logistics
  - Groups
  - Sharing & Points
  - Grading
- Project

Course Content & Goal

- **Presenting your work**
  - HTML, CSS, Drawing Software, LaTeX, Gnuplot
- **Code warrior**
  - Reading and Writing Code, Profiling, Debugging, Editors, IDE, Makefiles, Compilation, Linking, Version Control,C++, Java, Python, Bash, Awk
- **Linux**
- **Goal:** prepare you with vocabulary of CS tools such that later concepts can be easily explored
- **References:** on Moodle

Course: Instructor's take

- **Hone your programming expertise**
  - Develop a respect for programming
- **Engage more intimately with CS**
  - "Look beneath the hood" of software
  - Prepare yourselves for the cool stuff to be learnt

How do we achieve these goals?

Grading (8 credits)

- Assignments (9 labs): 45%
- Quiz (3) pre/post midsem 15%
  - (quiz in midsem slot)
- Project Initial Presentation 10%
- Project 30%
  - Including all work products
- **Total** **100%**
  (Scheme subject to variation –
  but you'll get adequate notice)

First Steps

- Form teams:
  - Pre-Midsem: Individual assignments
  - Post-Midsem: 3 ppl/assignments + Project
  - TAs will connect before Midsem to form teams
  - Details on Moodle
- Prior experience:
  - Query prior knowledge
  - Topics of your interest
- Project
  - Starts post-Midsem
  - Think about something cool

Submissions

- Lab Assignments due at 12noon Sunday
  - Via Moodle (If Moodle is down we'll give alternate)
- Missed deadline:
  - Submit assignment to prevent getting a fail grade
- Programming assignments:
  - Late submission (every 2-hour or part thereof) have an exponential penalty starting with 1%
  - Penalties:
  - 0-2(1%), 2-4(2%), 4-6(4%), 6-8(8%), 8-10(16%), 10-14(32%),14-24(64%),>24(100%) But you still have to submit it.

Grade Revision

- Bring to OUR attention within 72 hours of receipt or next lecture, whichever is earlier
- Post query so others affected can benefit
- Study model answers before questioning decision
- Appeal=> instructor may revise other marks also
- Be objective: ”Answer states 2 marks for this step; I have written this step; please reevaluate”

Course rules

- Interactive
- Strict on attendance
- Fussy about learning
- May not know too much
  but willing to work hard and do her job
- Logistics:
  - Moodle
  - Meetings w/TAs (Wed. 2-5pm) MS Teams channels
  - Course Web Page

Groups

- Groups of 3, fixed for the semester
- However, individual marks
- Lowest roll number submits
- Questions
  - Who does the work? Everyone!!
  - How to get help? Ask the TAs!
  - What help is permissible? References/resources!
  - What's the penalty for breaking rules?
    Worst case – grade reduction!

Important

Ethics

- **Important preparation for future**
- **Default Honour code** (write on assignment):
  - Pledge: “I’ve not given or received unauthorized assistance on this task”
- **Collaboration:**
  - Discuss with **own** group or TA
  - Inter group discussions not allowed.
- **Project:**
  - Feel free to take things from Internet but do not plagiarize (cite sources if you do)
  - Violation is a serious matter

Sharing & Points

- Any work for which you (or your group) claim points must be done by you
- If it is not your work, you must **explicitly** cite
- Cannot take or borrow something for which you claim points

Penalties

- What happens if you violate these?
- Don't --- its about learning and not points
- Will result in reporting to DDAC

Project

Project

- **Assessment:**
  - 40% of CS699 Course Assessment
- **Ideate:**
  - Consider yourself developers of a product/ service
- **Project process**:
  - ***Stages****:* Idea exploration; Articulation; Debugging; Prototype; Final Demo & presentation;
  - ***Submissions****:* presentations, working demo, documentation, video, documented code, reusable artifacts (code)
  - Work iteratively, incremental development!!
  - Use Github! Important!!

Summary

- **Hone programming skills & Linux OS**
  - Reading and Writing Code, Profiling, Debugging, Editors, IDE, Makefiles, Compilation, Linking, Version Control, Java, Python, Bash, Awk
- **Train you to present your work**
  - HTML, CSS, Drawing Software, LaTeX, Gnuplot
- **Project**
  - Brings it all together
- **Prepares you to get the most out of CS**