



INTERNATIONAL INSTITUTE OF
INFORMATION TECHNOLOGY

H Y D E R A B A D

Software Systems Development

Who are we?

Y. Raghu Reddy

Associate Professor,

Software Engineering Research Center

Research Interests: Software Engineering, Human Computer Interaction

Instructors:

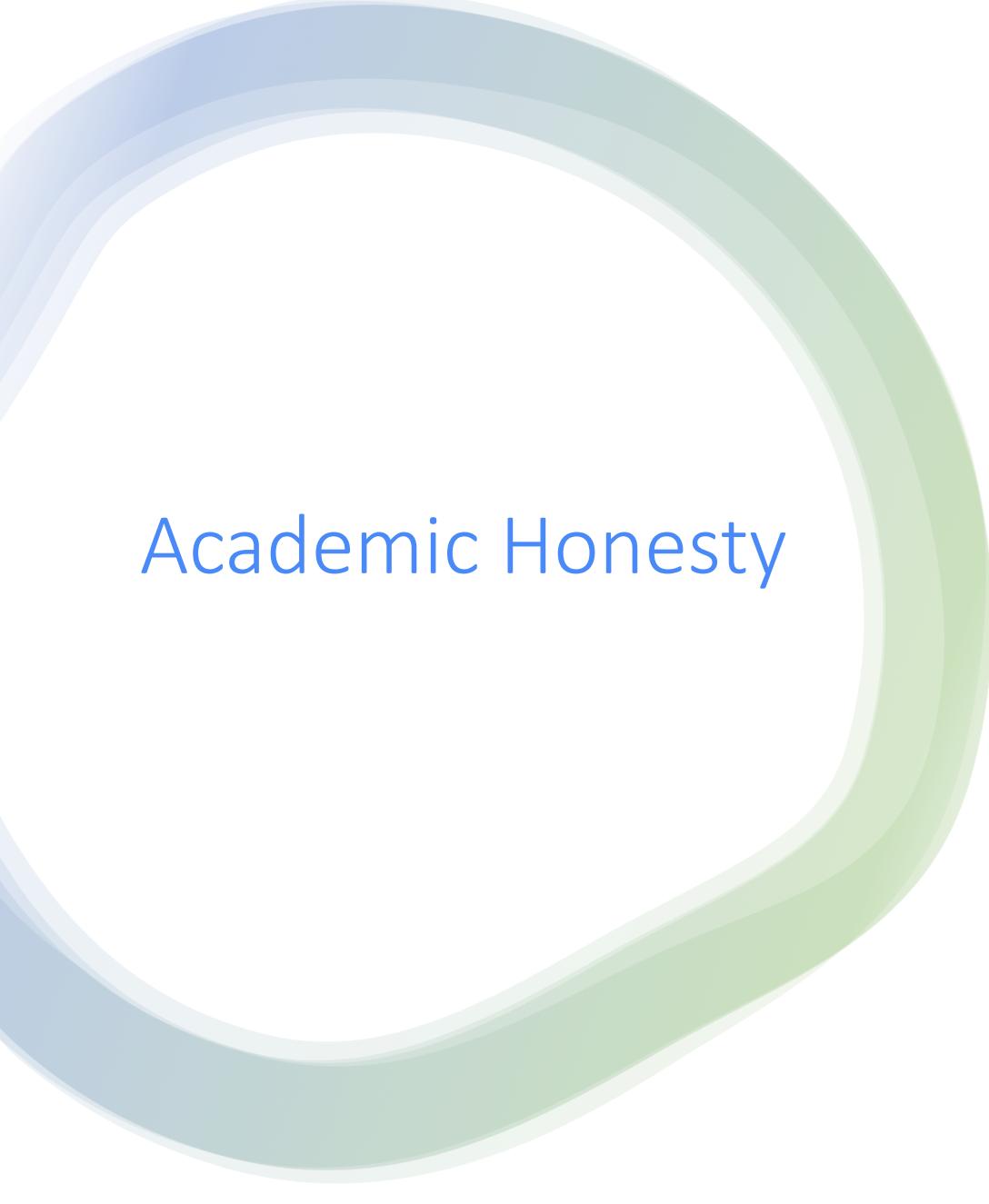
Lalit Mohan (Wells Fargo, IDRBT, PhD research scholar)

Sai Anirudh (SkillSoft, PhD research scholar)

Abhinav Gupta (Optum Healthcare, PhD research scholar)

Course Details

- The overall goal of this course is to provide a working knowledge on tools for building software systems.
- **Course Structure:** 2 Class per week (1 hour 25 min per class), Lab Work – 1 every week (2 hrs per week)
- **Grading Criteria:** Assignments – **40%**, Project – 30%; Quizzes/Lab Activities – **30%** (split to be decided later)
- **Course Notes:** Reference Material and relevant notes will be made available on Moodle. Students are expected to read the notes, put on effort, work towards rising your problem-solving skills and **learn things by doing**.
- **Facilitators:** Instructors and Teaching Assistants.
- **Time:** Honor Time and Come with learning mindset. Ensure that you record your queries and discuss them offline (using MS Teams chat or Moodle) when we run out of time.
- **Lab Work:** Linux Commands, Shell Scripting, HTML, CSS, AFrame, JavaScript, Python



Academic Honesty

A helps B in task X

⇒B doesn't get opportunity to do task X

⇒B doesn't learn the skill to do task X

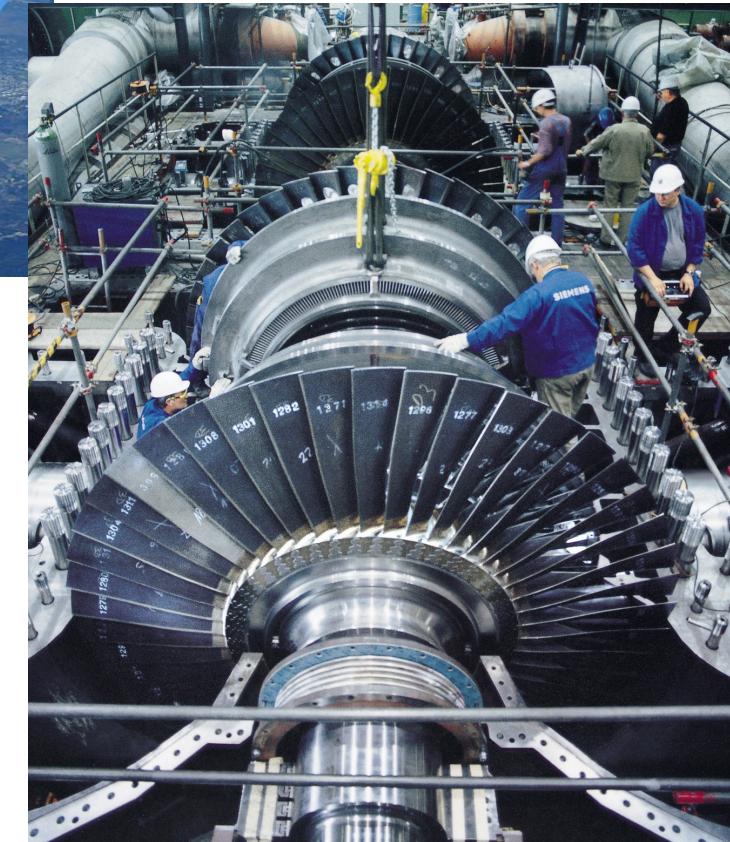
⇒B gets spoilt, dependent and unfit for jobs requiring skills of X

⇒You may think it is okay to do it only once and not repeat it. But when a thing is done once, it gets wired into the brain as being “okay”; and unless there is a strong reason, it *will* repeat.

If you want to help, help to learn.

What's common in these?

- Large complex “systems” with *lot* of software & hardware.
 - The Boeing 777 flies with over 4,000,000 lines of code on-board.
 - A typical top-level game has between 1 and 2 M SLOC (source lines of code)
 - Thousands of devices



What's a System?

- Commonly used/understood definition
 - Set of inter-related components working together to achieve a common objective
- A system may be “natural” or “Engineered”
 - Solar system (Natural)
 - Telephone network system, power plants, etc. (Engineered)
 - Systems have boundaries – due to various reasons

This course is about... Concepts, tools and technologies for developing Software Systems.

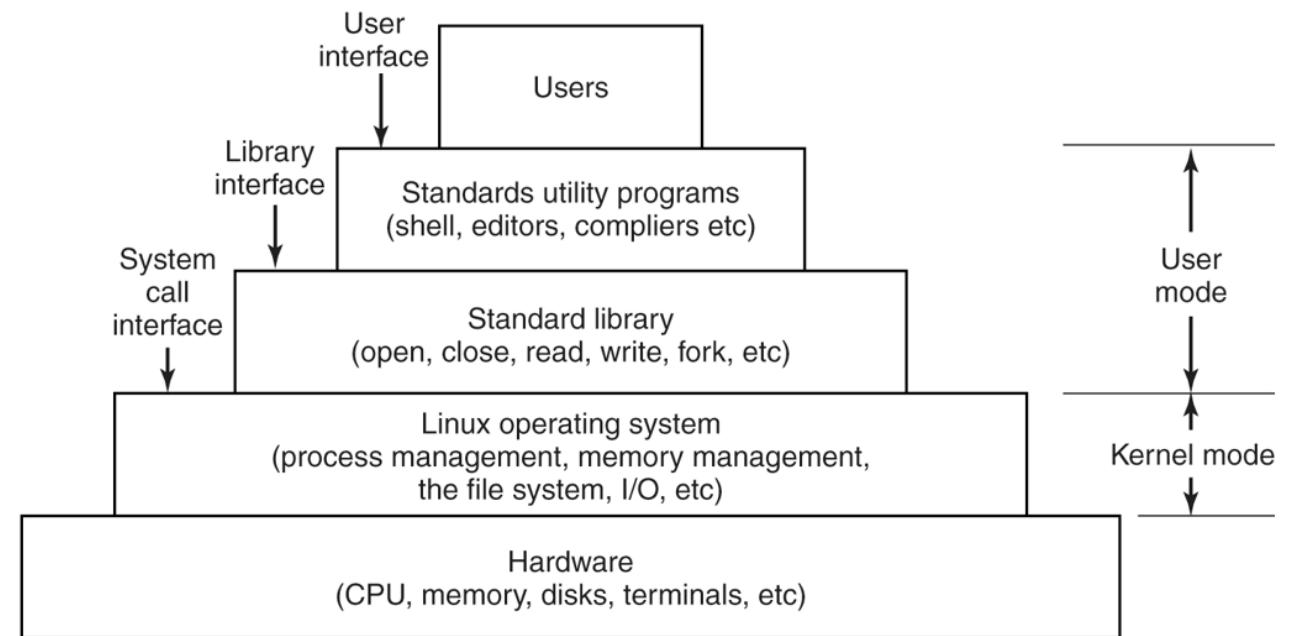


Broad set of topics...

	Basics of OS and Commands	Linux Intro and Basic Bash Commands
Shell Scripting	Shell Scripting	Advanced Linux Commands including file management and schedulers
	Shell Scripting	With examples on control flows, Continue with more examples, Regex, AWK and SED
Web Technologies - I	HTML	HTML advanced - DOM, XLST, DHTML, CSS, Web and App Servers
	Javascript	Javascript Intro, DataTypes, Variables, Operators, If-Else, Functions, Loops, Function Expressions, Events, Form Controls, Regular Expressions
	Javascript	JavaScript Libraries
	A-Frame	WebVR, Simple AR apps
Web Technologies - II	Python	Functions, Exceptions and Error Handling, Sequences, falsy values, scoping rules, closures, high order functions, mutability, object model and classical inheritance, multiple inheritance and MRO, modules and packages, variable args, decorators
	Python	File Operations and usage of libraries including SOAP and REST API
	Python	Simple Flask based server with AJAX and CORS enabled, and other libraries
Systems Engineering	SDLC	SDLC, Version Control, Github, Editors, Software Architectures, Bugs and Bugtrackers, AST, Flavors of Programming Languages, OOPS, MVC and Design Patterns, Functional and Non Functional (Security, Performance and Usability)
	DB	Normalization rules, DDL, DML, DCL, CRUD, Basics of SQL, Syntax, Coding Styles, Lint Tools
	Network	Computer Networks
	Security	Information and Cyber Security

Why Linux

- Free
- Open Source
- Highly Secure
- Runs Fast
- Several Distros – almost 600+
 - Ubuntu, CentOS, Debian, Elementary OS, Linux Mint, Tails, Rocks cluster Distribution, etc.



SHELL - A program (a.k.a. command-line interpreter) that allows the user to interact with the UNIX/Linux system.

Examples: Bourne shell (sh), Bourne again shell (Bash), C shell (csh, tcsh), Korn shell (ksh), Powershell (windows)

