



INTERNATIONAL INSTITUTE OF
INFORMATION TECHNOLOGY

HYDERABAD



Introduction to Software Systems

Spring 2021 - Term 3

Who are we?

Y. Raghu Reddy

Software Engineering Research Center

Co-Instructors:

Sai Anirudh (SkillSoft, PhD research scholar)

Abhinav Gupta (Optum, PhD research scholar)

Course Details

- **Course Objective:** The aim of this course is to provide a working knowledge on tools and processes for building simple software systems.
- **Course Structure:** 16 classes (1 hr per class), Lab Work – 3 hrs per week (7 lab sessions overall)
- **Grading split up:** Quiz – 15%, Final Exam – 20%, Assignments – 25 % (3), Labs – 20% (4); Others– 20% (Activities, Surprise quiz/test, Inclass Activity, etc)
- **Course Notes:** Reference Material and relevant notes will be made available on Moodle and on Github. Students are expected to read the notes/reading material, put on effort, work towards rising your problem-solving skills and learn things by doing.
- **Lab Work:** Linux Commands, Shell Scripting, HTML, CSS, JavaScript, Python
- **Books/Materials:**
 - Mastering Linux Shell Scripting : A practical guide to Linux command-line, Bash scripting, and Shell programming, by Mokhtar Ebrahim, Andrew Mallett
 - Learning Python: Powerful Object-Oriented Programming, by Mark Lutz
 - JavaScript: The Definitive Guide, by David Flanagan
 - Software Engineering Principles (from various sources)
 - Workbook given by the course instructors
 - <https://serciiit.gitbook.io/introduction-to-software-systems/>

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 spoiled, 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.

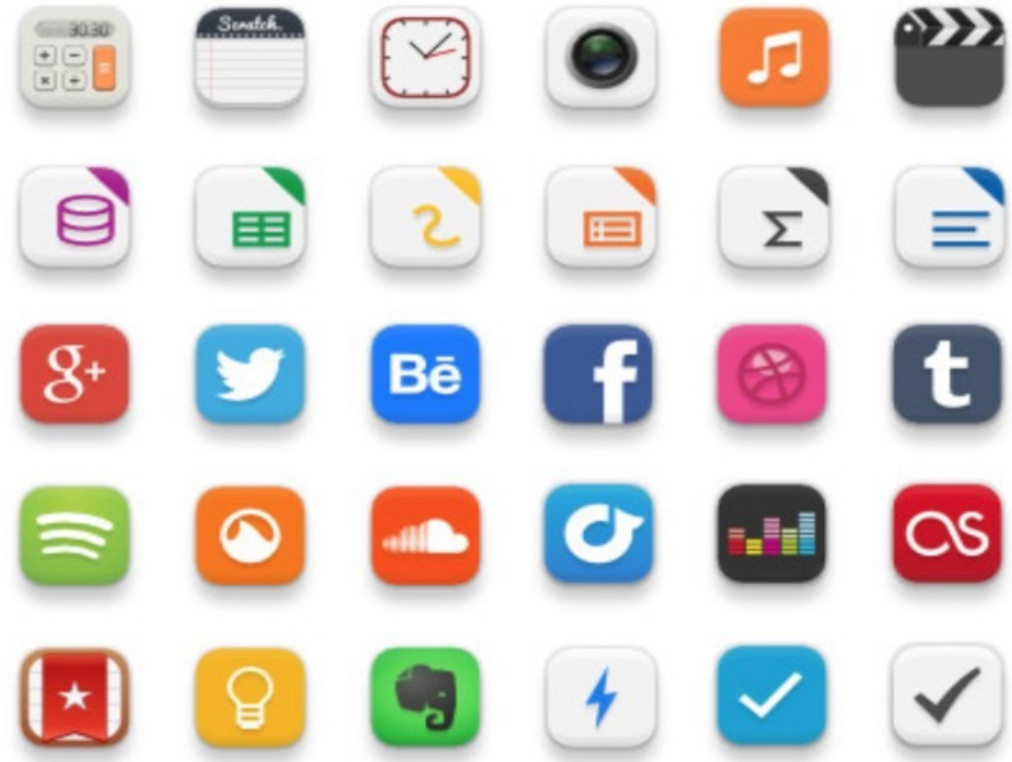
If you want to help, help to learn.

What's a System?

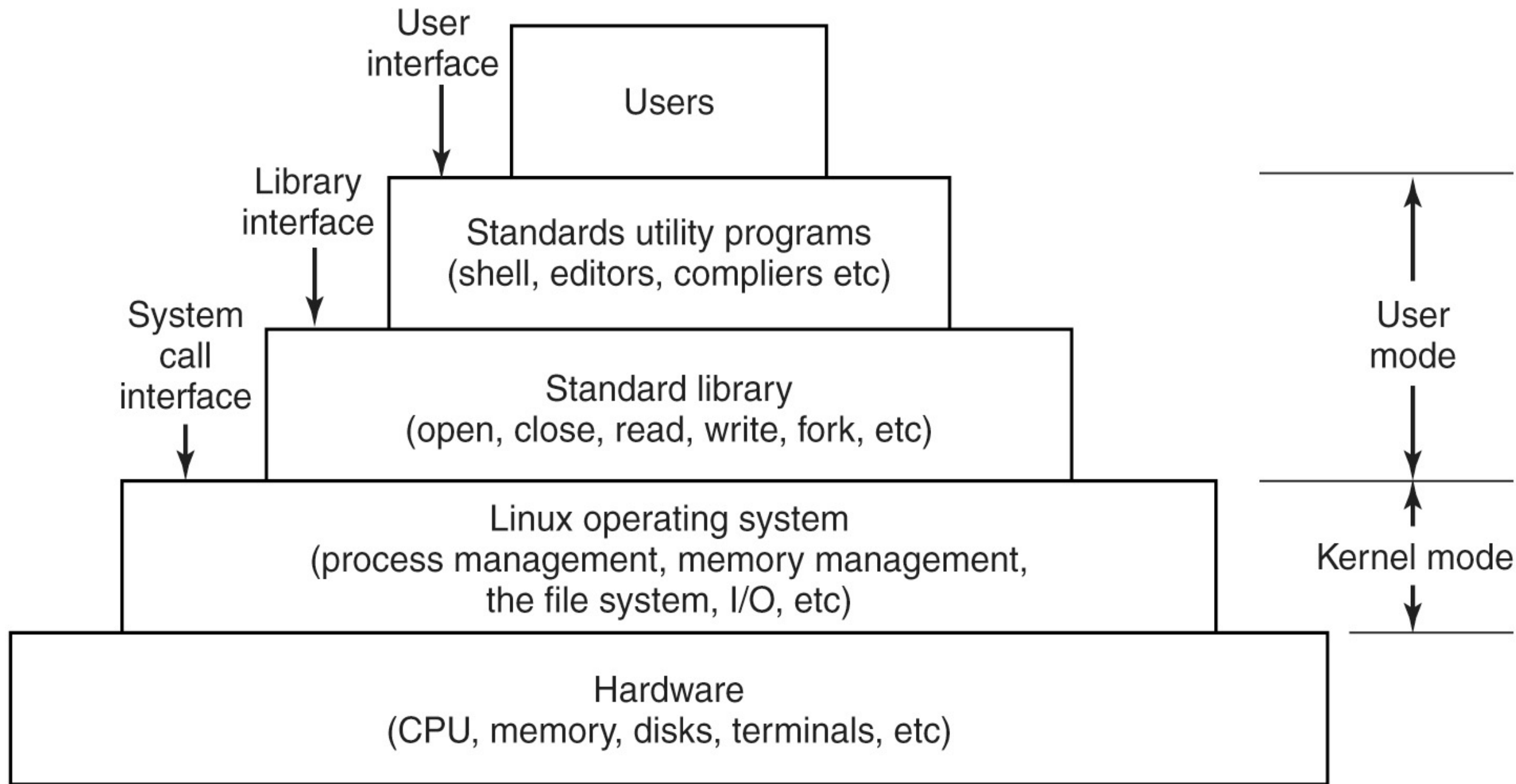
- Commonly used/understood definition
 - Set of inter-related components working together to achieve a common objective
- A system may be “Natural” or “Engineered”
 - Solar system (Natural)
 - Telephone network system, power plants, etc. (Engineered)
 - Systems have boundaries – due to various reasons



Hardware

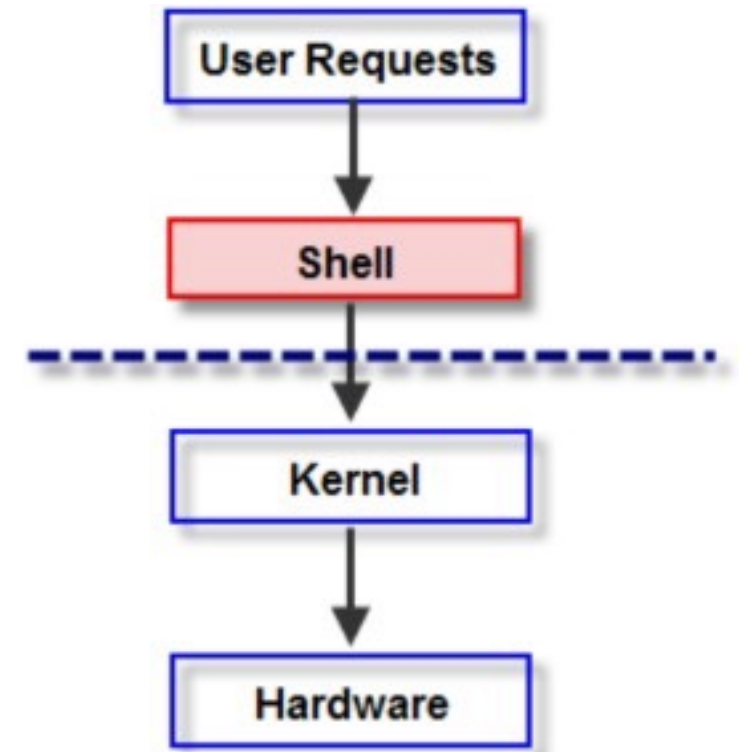
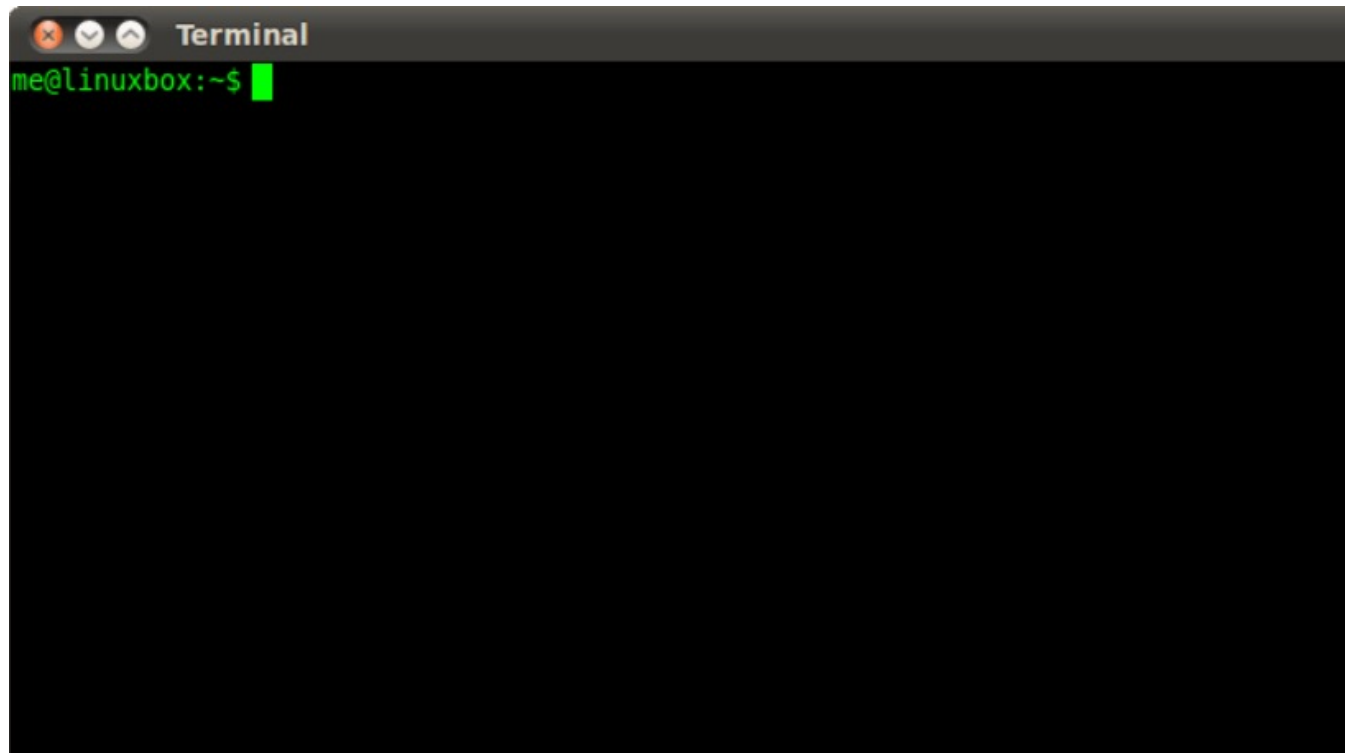


Software



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

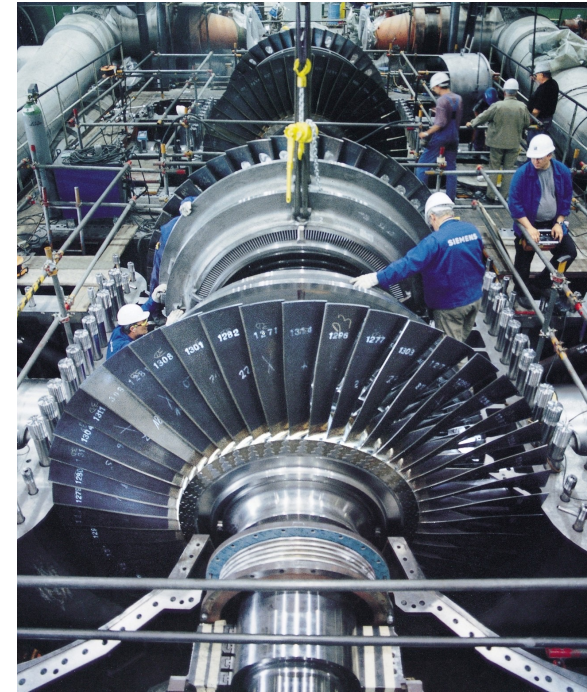
Examples: Bourne shell (sh), Bourne again shell (Bash), C shell (csh, tcsh), Korn shell (ksh), Powershell (windows)



What's common in these?

They are 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



Programs

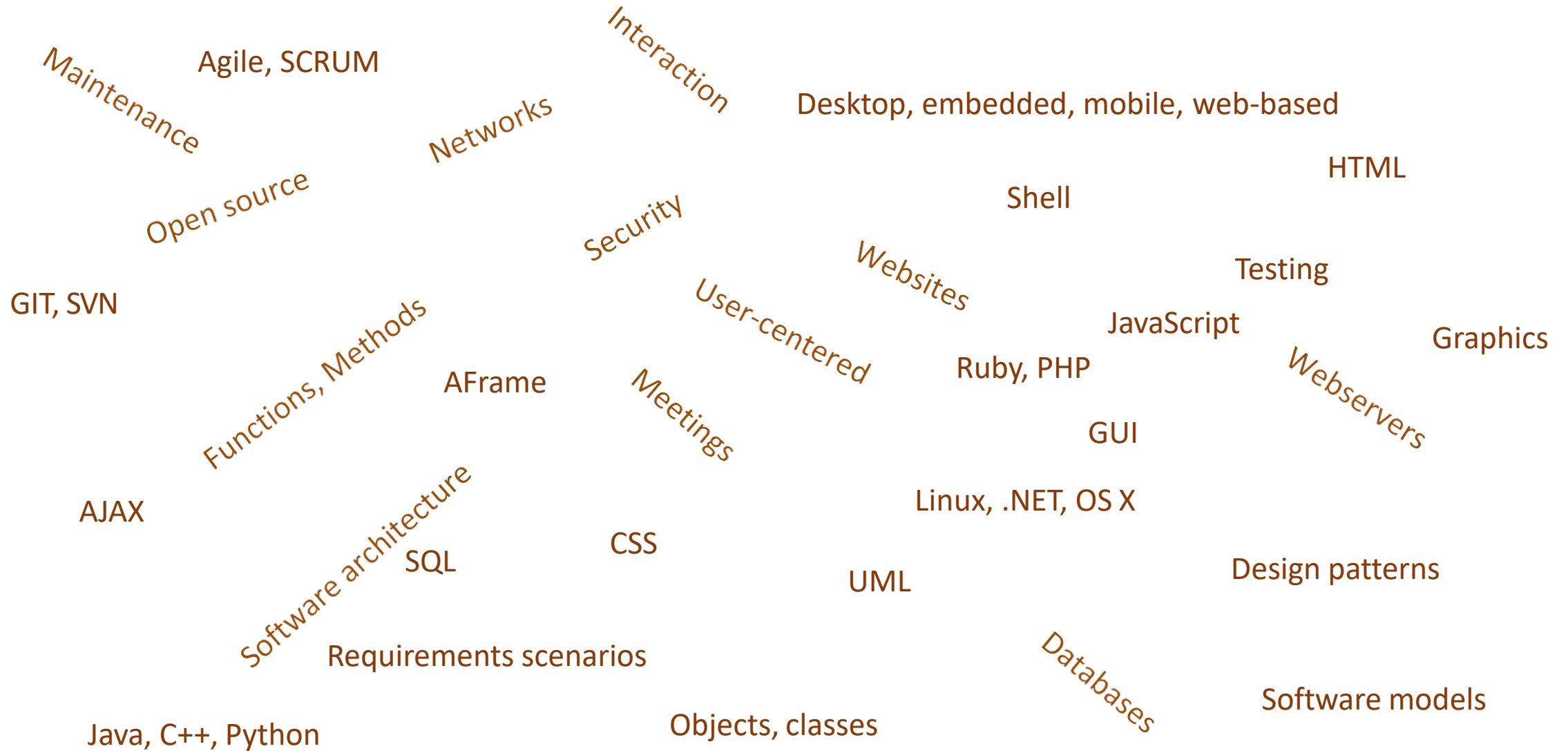
Teamwork

Process

Engineering design

Communication

This course is about... Tools and Technologies for Software Systems



Keep checking MOODLE !!!