

**Foundations of Modern Software Systems.**  
**Elective Course for 5th Semester Students**  
**Instructor - Piyush Goel**

**Course Introduction** - In present times, our day-to-day lives are governed by software. All our daily activities such as ordering groceries, searching for doctors, and communicating with our loved ones are enabled by handheld devices like cell phones, tablets, laptops, and desktops. The apps that we use daily are powered by some core principles of Computer Science, and Software engineering.

This course aims to give the students a peek into what goes behind the scenes of modern-day applications. The course will use a case study of a food delivery application like Swiggy/Zomato, and dissect the layers on which such apps are built. The suggested course break-up is given below.

**1. Module-1: Introduction (3 hours)**

- a. Overview
  - i. Overview of a food delivery business such as Swiggy, Zomato.
  - ii. Different Components of the software powering such a business.
- b. Foundations of Software Systems
  - i. Storage, Compute, Networking
  - ii. Turtles All the way down!

**2. Module-2: Storage (12 hours)**

- a. Data Types
- b. Fundamentals of data storage (File systems, Serialisation, Compression, etc.)
- c. Structuring and Querying Data.
- d. Databases
  - i. Relational
  - ii. Non-Relational
    - 1. Key-Value
    - 2. Document
    - 3. Time-Series
- e. Object Storage (Files, and Blobs)
- f. Brief about advanced concepts such as CAP Theorem (Optional)
- g. Practicals - Experiment, and play around with the various databases.

**3. Module 3: Compute (8 hours)**

- a. Process, Threads, Memory.
- b. Servers, Virtual Machines, and Containers.
- c. Application Servers (Web Servers, FTP, Email)
- d. Cloud Computing Basics.
- e. Practicals - Playing with Cloud providers such as AWS.

#### **4. Module-4: Networking (12 hours)**

- a. Building Blocks (OSI Layer Models, Network Topologies, Subnets, VPNs, etc.)
- b. Naming Systems (DNS)
- c. Important Protocols
  - i. TCP/UDP
  - ii. HTTP
  - iii. Web Sockets
- d. Network Security (SSL/TLS)
- e. Practicals - Intercepting Network Traffic and analyzing it in detail. Basic commands for tinkering with computer networks.

#### **5. Module-5: Assignment (6 hours) – Optional**

- a. Building a minified version of a real-life application. The students will be expected to work on the assignment in their free time and show the progress to the instructor on a weekly basis. The assignment shall be done in groups.

**Mode of Instruction:** Online, and Offline classes shall be conducted. The instructor might create a Telegram or an email group to engage the students in group discussions, and knowledge-sharing sessions.

**Study Material:** The relevant books, papers, and blogs will be recommended by the instructor when commencing each module.

#### **Pre-Requisites for the students:**

1. Access to Unix-based laptops, or desktops.
2. Basic command over a scripting language such as Python, Ruby, PHP.
3. Most of all - Curious minds, and a deep hunger to learn. =)

#### **Instructor:**

Piyush Goel (aka, Goel) is the former SVP of Engineering at Capillary Technologies, an industry leader in Consumer Engagement software. Goel has over 15 years of industry experience with stints at companies such as Capillary, InMobi Technologies, Travel Triangle Pvt Ltd, and Yahoo Inc. He holds a BTech, and MTech in Computer Science, and Engineering from the Indian Institute of Technology, Kharagpur, and has over 8 publications in International conferences and Journals. He is passionate about building software systems that impact human lives and loves to teach and mentor young engineers in his spare time. He likes to travel and is on a spiritual pursuit via the Buddhist meditation practice of Vipassana.

LinkedIn - <https://www.linkedin.com/in/piyushgoel1/>

Twitter - <https://twitter.com/pigol1>

bQyHhQLzd77-GbGYoBYCHcab\_YANZws4zmuNM5gkj7mu3rgErlvmOZG-lqp5C0rEHWy-zGv44qCW  
e1TMbxLojA==

bucket="test"

write\_api = write\_client.write\_api(write\_options=SYNCHRONOUS)

```
for value in range(5):
    point = (
        Point("measurement1")
        .tag("tagname1", "tagvalue1")
        .field("field1", value)
    )
    write_api.write(bucket=bucket, org="test", record=point)
    time.sleep(1) # separate points by 1 second
```

query\_api = write\_client.query\_api()

```
query = """from(bucket: "test")
|> range(start: -10m)
|> filter(fn: (r) => r._measurement == "measurement1")"""
tables = query_api.query(query, org="test")
```

```
for table in tables:
    for record in table.records:
        print(record)
```