# Lecture 02 (In Order Pipelines)

## 1 5 Stage Pipeline (recap)

1. Instruction fetch - IF
2. Instruction decode (+ operand fetch) - OF
3. Execute stage - EX
4. Memory access - MA
5. Register write-back - RW

This is part of GATE syllabus and we are *GATEd*

There are back connections from:

1. EX to IF - for branching
2. RW to OF - for write back

## 2 In-Order Pipelining

1. Multiple instructions were at different places in the pipeline
2. This leads to issues:
   i. Structural hazards
   ii. Instruction dependency
      - insert nop aka bubbles
      - forward the value using multiplexers
      - forwarding paths:
        1. RW → MA
        2. RW → EX
        3. RW → OF
        4. MA → EX
      - EPIC project tried to move this to compiler stage but didn't go anywhere
      - Load use hazard leads to impossibility when EX needs MA load
   iii. Control hazard when branching
      - predict
      - execute branch independent instructions (compiler level optimisation) - delayed branch
      - need to balance between compiler and hardware optimisations

(kids were running + trekking on stairs and not losing breath even when talking)