## TOPIC NO: 35

## TOPIC NAME: Implement a student course registration system where students can select/drop courses and calculate credit hours

## GROUP NAME: SYNTAX SQUAD

## MEMBERS:

1. MIFRAH IQBAL
2. SADIA
3. SANA

# Project Process & Strategies

## Student Course Registration System

## 1. Project Objective

The goal of this project is to design and implement a Student Course Registration System where students can:
- Register (add) for courses
- Drop courses
- View their current registered courses
- Calculate total credit hours

## 2. System Requirements

### Functional Requirements

1. Students can view available courses with their credit hours.
2. Students can add/select courses (must not exceed the maximum credit hours, e.g., 18).
3. Students can drop courses.
4. The system calculates the total registered credit hours.
5. Students can view registered courses at any time.

### Non-Functional Requirements

- User-friendly interface (CLI or GUI).
- Error handling (e.g., dropping a non-registered course, exceeding max credits).
- Scalability (expandable to multiple students, database/file support later).

## 3. System Design Strategy

### a) Data Structures

- Available Courses: Stored in a dictionary (CourseCode → {CourseName, CreditHours}).
- Registered Courses: Stored in a list/dictionary for each student.

Example:

CS101 → Intro to Programming, 3 Credits
MATH201 → Calculus II, 4 Credits
ENG105 → English Composition, 2 Credits
PHY110 → Physics I, 3 Credits

### b) Core Functions

1. view_courses() → Show all available courses
2. register_course(course_code) → Add a course (check max credits)
3. drop_course(course_code) → Remove a course
4. calculate_credits() → Calculate total registered credits
5. view_registered_courses() → Display student's registered courses

## 4. Process Workflow

1. Display menu options:
   - View available courses
   - Register a course
   - Drop a course
   - View registered courses
   - Calculate total credit hours
   - Exit

2. Student selects an option.
3. Perform the action with error checks.
4. Loop back to the menu until exit.

## 5. Project Strategies

- Phase 1: Build a CLI-based prototype in Python.
- Phase 2: Add validation (max credits, duplicate registration).
- Phase 3: Store data in files (JSON/CSV) for persistence.
- Phase 4 (Optional): Add GUI (Tkinter / Web app with Flask or Django).

## 6. Workflow Visualization