

### กิจกรรมที่ 3 : Encapsulation & Refactoring

1. บริษัทได้รับมอบหมายให้ปรับปรุงระบบทะเบียนนักศึกษาเก่าที่เขียนด้วยภาษา C (Legacy Code) ซึ่งมีความยุ่งยากในการจัดการหน่วยความจำและข้อมูลกระจัดกระจาย ท่านได้รับหน้าที่ให้เขียนระบบใหม่ ด้วยภาษา Python โดยใช้หลักการ Object-Oriented Programming (OOP) และ Encapsulation อย่างเคร่งครัด
2. ศึกษา Logic การทำงานจากโค้ดภาษา C (โดยเฉพาะการคำนวนเกรด และการตรวจสอบการลงทะเบียนชั้น)
3. ข้อกำหนด (Requirements) ให้นักศึกษาออกแบบ Class ใหม่ใน Python โดยต้องปฏิบัติตามกฎดังนี้
  - **Class Design:** สร้าง Class โดยกำหนด attribute ทุกด้วยให้เป็น **Private** (`_variable`)
  - **Encapsulation Rules:**
    - ห้ามเข้าถึง attribute โดยตรงจากภายนอก Class
    - สร้าง Getter/Setter เพื่อที่จะเป็นเท่านั้น (หักคะแนนหากสร้างพร้อมเพรียง)
    - ห้ามใช้ Global List/Dictionary ในการเก็บข้อมูลหลัก ข้อมูลต้องถูกเก็บอยู่ภายใน Instance ของ Class เท่านั้น
4. **Refactoring Logic (C → Python Method):** ต้องย้าย Logic จาก Global Function ในภาษา C เข้าไปเป็น Method ใน Class ดังนี้:
  - `enroll_to_subject(std, sub)` → `Student.enroll(subject)` ลงทะเบียนเรียน (คืนค่า "Done", "Already Enrolled", "Error")
  - `drop_from_subject(std, sub)` → `Student.drop(subject)` ถอนรายวิชา (คืนค่า "Done", "Not Found", "Error")
  - `assign_grade(std, sub, grade)` → `Student.assign_grade(subject, grade)` บันทึกเกรด (A=4, B=3, C=2, D=1, F=0)
  - `get_student_GPS(std)` → `Student.get_gps()` คำนวนเกรดเฉลี่ย (คืนค่า 0.0 หากไม่มีหน่วยกิต)
  - `search_subject_that_student_enrolled` → `Student.get_enrolled_subjects()` คืนค่าเป็น List ของรายวิชาที่ลงทะเบียนอยู่
5. กำหนดให้แต่ละรายวิชา มีผู้สอน 1 คน และ นักเรียนสามารถลงทะเบียนในรายวิชาได้ ก็ต่อเมื่อจำนวนนักเรียนที่ลงทะเบียนในรายวิชาได้ และนักเรียนสามารถถอนรายวิชาได้ หากไม่ต้องการเรียนอีกต่อไป
6. หลังจากการเรียน จะมีการสอบและให้เกรด โดยจะต้องเก็บข้อมูลว่า นักเรียนแต่ละคนได้เกรดในรายวิชา ได้เป็นเกรดใด กำหนดให้ A=4, B=3, C=2, D=1, F=0
7. ให้นักศึกษาเขียนโค้ดลงในไฟล์ `lab3_skeleton.py`

## 8. Test Case จะมีการทำงานดังนี้

- ก่อนเริ่มการทดสอบ ระบบจะทำการสร้างข้อมูลเบื้องต้น (Setup Data) ได้แก่:
  - สร้าง Instance ของ Teacher จำนวน 1 ท่าน
  - สร้าง Instance ของ Subject จำนวน 3 วิชา และทำการ Assign Teacher ให้เรียบร้อย
  - สร้าง Instance ของ Student จำนวน 2 คน
- งานนี้จะทำการทดสอบตามลำดับดังนี้:
  - **Test Case 1:** Testing Enrollment Logic (ทดสอบระบบลงทะเบียน)
    - ทดสอบการลงทะเบียนปกติ: สั่งให้นักศึกษาลงทะเบียนวิชาที่ 1 ต้องได้ผลลัพธ์ "Done"
    - ทดสอบการลงทะเบียนซ้ำ: สั่งให้นักศึกษาคนเดิมลงทะเบียนวิชาเดิมซ้ำ ต้องได้ผลลัพธ์ "Already Enrolled"
    - ทดสอบลงทะเบียนวิชาเพิ่ม: ลงทะเบียนวิชาที่ 2 ต้องได้ผลลัพธ์ "Done"
  - **Test Case 2:** Testing Drop Logic (ทดสอบระบบถอนรายวิชา)
    - สั่งให้นักศึกษาถอนรายวิชาที่ 2 ต้องได้ผลลัพธ์ "Done"
    - ตรวจสอบความถูกต้องของข้อมูล: เรียก method get\_enrolled\_subjects() เพื่อยืนยันว่าเหลือนักศึกษาลงทะเบียนอยู่เพียง 1 วิชาเท่านั้น
  - **Test Case 3:** Testing GPA Calculation (ทดสอบการคำนวณเกรดเฉลี่ย)
    - ให้นักศึกษาคนที่ 2 ลงทะเบียนครบ 3 วิชา (หน่วยกิต 3, 3, และ 4 ตามลำดับ)
    - ทำการตัดเกรด A, B, และ C ตามลำดับ
    - คำนวณเกรดเฉลี่ย (GPA) ผลลัพธ์ที่ได้ต้องถูกต้องตามสูตรคำนวณ (Expected: 2.90)
  - **Test Case 4:** Testing Grade Update (ทดสอบการแก้ไขเกรด)
    - ทำการแก้ไขเกรดของวิชาที่ 2 จากเดิมได้ B เป็น A
    - เรียกดูเกรดเฉลี่ยใหม่อีกครั้ง ค่า GPA ต้องเปลี่ยนแปลงตามทันที (Expected: 3.20)
  - **Test Case 5:** Testing Edge Cases (ทดสอบกรณีพิเศษ)
    - สร้างนักศึกษาใหม่ที่ยังไม่ลงทะเบียนเรียน แล้วเรียกดู GPA ต้องได้ค่า 0.0 (หมาย Error)
    - ให้นักศึกษาคนเดิมลงทะเบียนเรียน แต่ยังไม่ถูกตัดเกรด แล้วเรียกดู GPA ต้องได้ค่า 0.0
  - **Test Case 6:** Testing Robustness (ทดสอบความทนทานของโปรแกรม)
    - ทดสอบล่งข้อมูลผิดประเภท (Invalid Type) เข้าสู่ระบบ เช่น ส่งค่า String เข้าไปใน method enroll() แทนที่จะเป็น Subject Instance
    - โปรแกรมจะต้องคืนค่า "Error" หรือแจ้งเตือนอย่างเหมาะสม โดย หมายทำให้โปรแกรมหยุดทำงาน (Crash)

- **Test Case 7:** Testing Logic Consistency (ทดสอบการทำการรายการกับวิชาที่ไม่ได้ลงทะเบียน)
  - Drop Non-enrolled: ทดสอบลั่งถอนรายวิชาที่นักศึกษายังไม่เคยลงทะเบียน  
Expected Result: โปรแกรมต้องคืนค่า "Not Found" (เนื่องจากไม่มีข้อมูลให้ลบ)
  - Grade Non-enrolled: ทดสอบบันทึกเกรดในรายวิชาที่นักศึกษายังไม่เคยลงทะเบียน  
Expected Result: โปรแกรมต้องคืนค่า "Not Found" หรือ "Error" (เนื่องจากหา Enrollment ไม่เจอ)