

กิจกรรมที่ 2 : Encapsulation

1. สร้างคลาสต่อไปนี โดยใช้หลัก Encapsulation ซึ่งจะไม่ยอมให้มีการเข้าถึงข้อมูลโดยตรง โดยให้กำหนด attribute ทุกตัวเป็นแบบ private
 - Student โดยมี attribute : student_id, student_name
 - Subject โดยมี attribute : subject_id, subject_name, credit
 - Teacher โดยมี attribute : teacher_id, teacher_name
 - ให้สร้าง setter, getter เท่าที่จำเป็น คือ สร้างเฉพาะเมื่อมีการร้องขอ หรือ กำหนดค่าจาก ภายนอกเท่านั้น ถ้าพบ setter, getter ที่ไม่จำเป็น จะหักคะแนน
 - สามารถสร้าง class อื่นๆ หรือเพิ่ม attribute ได้ตามความเหมาะสม (แนะนำ : มีเพิ่ม 1 class)
2. กำหนดให้แต่ละรายวิชา มีผู้สอน 1 คน และ นักเรียนสามารถลงเรียนในรายวิชาใดๆ ก็ได้ จำนวนกี่วิชาก็ได้ และนักเรียน สามารถถอนวิชาได้ หากไม่ต้องการเรียนอีกต่อไป
3. หลังจากการเรียน จะมีการสอบและให้เกรด โดยจะต้องเก็บข้อมูลว่านักเรียนแต่ละคนได้เกรดในวิชาใด เป็นเกรดใด กำหนดให้ A=4, B=3, C=2, D=1, F=0
4. ให้สร้าง method ที่ทำงานดังนี้
 - method สำหรับลงทะเบียน (enroll_to_subject) โดยส่ง argument เป็น instance ของ Student และ Subject หากนักเรียนยังไม่ลงทะเบียนวิชานั้น ให้เพิ่ม object **การลงทะเบียน** และ return "Done" หากลงแล้วให้ return "Already Enrolled" แต่ถ้า argument ไม่ถูกต้องให้ return "Error"
 - method สำหรับถอนรายวิชา (drop_from_subject) โดยส่ง argument เป็น instance ของ Student และ Subject หากพบการลงทะเบียนให้ ดำเนินการถอน (ลบข้อมูลการลงทะเบียน) และ return "Done" หากยังไม่ได้ลงทะเบียนให้คืนค่า "Not Found" แต่ถ้า argument ไม่ถูกต้องให้ return "Error"
 - method สำหรับค้นหา**การลงทะเบียน**ในรายวิชา (search_student_enrolled_in_subject) โดยส่ง argument เป็น instance ของ Subject และ ส่งคืนมาเป็น List ของ instance ของ **การลงทะเบียน** ที่ลงในวิชานั้น ถ้าไม่มีให้คืนเป็น List ว่าง แต่ถ้า argument ไม่ถูกต้องให้ return "Error"
 - method สำหรับบอกจำนวนนักศึกษาที่ลงในวิชา (get_no_student_enrolled) โดยส่ง argument เป็น instance ของ Subject และคืนค่าเป็นจำนวน หากไม่เจอ ให้คืนค่า "Not Found"
 - method สำหรับค้นหา**การลงทะเบียน**ที่ นศ. ลงทะเบียน (search_subject_that_student_enrolled) โดยส่ง argument เป็น instance ของ Student และส่งคืนเป็น List ของ instance ของ **การลงทะเบียน** หากไม่เจอ ให้คืนค่า "Not Found"
 - method สำหรับค้นหาอาจารย์ที่สอนในวิชานั้น (get_teacher_teach) โดยส่ง argument เป็น instance ของ Subject และ ส่งคืนเป็น Instance ของอาจารย์ที่สอนวิชานั้น หากไม่เจอ ให้คืนค่า "Not Found"

- method สำหรับค้นหาการลงทะเบียนที่ นศ. ลงทะเบียนในรายวิชานั้น (search_enrollment_subject_student) argument เป็น instance ของ Subject และ Student และส่งคืนเป็น instance การลงทะเบียน (สำหรับเรียกใช้โดย assign_grade)
 - method สำหรับกำหนดเกรดให้กับนักศึกษาในรายวิชา (assign_grade) โดยส่ง argument เป็น instance ของ Student และ Subject และ เกรด (เป็น string) หากยังไม่มีเกรดให้ เพิ่ม เกรด ลงใน **การลงทะเบียน** และ return “Done” หากมีเกรดแล้วให้ return “Error” หากไม่พบให้คืนค่า “Not Found”
 - method สำหรับค้นหาผลการเรียนของนักศึกษา (get_student_record) โดยส่ง argument เป็น instance ของ Student และส่งคืนเป็น dictionary { 'subject_id' : ['subject_name', 'grade'] หากไม่เจอ ให้คืนเป็น dictionary ว่าง
 - method สำหรับคำนวณเกรดเฉลี่ยของนักศึกษา (get_student_GPS) โดยส่ง argument เป็น instance ของ Student และคืนเป็นเกรดเฉลี่ย ให้เรียกใช้ get_student_record
5. โปรแกรมจะต้องเป็นไปตามข้อกำหนดดังนี้
- ห้ามใช้ dictionary ในการเก็บข้อมูล
 - ห้ามเก็บข้อมูลนอกคลาส ยกเว้นกรณีที่เป็น list ของ object
 - ในการสร้างคลาสให้กำหนดว่าจะเก็บข้อมูลใด และ ห้ามมิให้เก็บข้อมูลที่ไม่ใช่ข้อมูลที่เกี่ยวข้องกับคลาสนั้น หากไม่สามารถเก็บลงในคลาสใดได้เลย ให้พิจารณาสร้างคลาสใหม่
 - ในคลาสไม่ให้มีการ Input ค่าหรือ print ค่าโดยตรง ให้ส่งข้อมูล parameter เข้าไปและได้ข้อมูลกลับมาเท่านั้น (ให้มอง class เป็น service)
 - ข้อมูลที่เก็บในคลาสที่ไม่ใช่คลาสพื้นฐาน เช่น คลาสนักศึกษา จะต้องเก็บข้อมูลเป็น Instance ของคลาสพื้นฐานเท่านั้น
6. Test Case จะมีการทำงานดังนี้
- สร้าง instance Student 10 คน
 - สร้าง instance Subject 3 วิชา
 - สร้าง instance Teacher 3 คน โดย add ชื่อผู้สอนเข้าไปในแต่ละวิชาที่สร้าง โดยใช้ method assign_teacher ของ instance วิชา
 - จะมีการเรียกใช้ function enroll_to_subject เพื่อลงทะเบียนนักศึกษาในแต่ละรายวิชา
 - Test Case #1 : กรณีที่ นศกับวิชานั้น ยังไม่เคยมีการลงทะเบียน การลงทะเบียนสำเร็จ ให้ return “Done” จากนั้นจะมีการเรียกใช้ function search_subject_that_student_enrolled(student_id) เพื่อตรวจสอบว่าคืนเป็น List ของวิชาที่ลงทะเบียนหรือไม่
 - Test Case #2 : ทดสอบ enroll_to_subject ในกรณีที่ argument ไม่เป็น instance
 - Test Case #3 : ทดสอบ enroll_to_subject ในกรณีที่มีการลงทะเบียนซ้ำ
 - Test Case #4 : ทดสอบ drop_from_subject ในกรณีที่ argument ไม่เป็น instance
 - Test Case #5 : ทดสอบ drop_from_subject ในกรณีไม่พบการลงทะเบียน
 - Test Case #6 : ทดสอบ drop_from_subject กรณี drop สำเร็จ

- Test Case #7 : ทดสอบ search_student_enroll_in_subject โดยกำหนดให้ instance ของ **การลงทะเบียน** ให้ทำ getter สำหรับ instance Subject และ instance Student และทำ getter สำหรับ student id สำหรับ class Student และ subject_id สำหรับ class Subject
- Test case #8 ทดสอบ get_no_of_student_enrolled โดยคืนค่าจำนวนนักศึกษา
- Test Case #9 ทดสอบ search_subject_that_student_enrolled ค้นหาวิชาที่นักศึกษาลงทะเบียน
- Test Case #10 ทดสอบ get_teacher_teach ค้นหา instance ของอาจารย์ที่สอนในวิชา
- Test Case #11 ทดสอบ search_enrollment_subject_student ค้นหา instance ของ **การลงทะเบียน** ของ นักศึกษา กับ รายวิชา
- Test Case #12 ทดสอบ assign_grade
- Test Case #13 ทดสอบ get_student_record
- Test Case #14 ทดสอบ get_student_GPS