

## Assignment 4 : White-Box Testing II

### 01518573 Software Testing and Verification

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยเกษตรศาสตร์

สำหรับงานแต่ละข้อด้านล่าง ให้ส่งงานโดยให้บีบอัดไฟล์และส่งงานตามที่ระบุในแต่ละข้อ

#### ข้อ 1 : ทดสอบด้วยเทคนิค Decision และ Condition Coverage ต่าง ๆ

พิจารณาโปรแกรมการล็อกอิน โดยมีการตรวจสอบเงื่อนไขว่า หากมีการล็อกอินที่ไม่ถูกต้องเกิน 3 ครั้ง จะล๊อคบัญชี รหัสคำสั่งในการตรวจสอบเงื่อนไขมีดังนี้

```
if ( (!validUser || !validPasswd) && count > 3) {  
    // .... โค้ดแสดงข้อความการล๊อคบัญชี ...  
}
```

จงออกแบบกรณีทดสอบด้วยเทคนิค structure-based ต่อไปนี้ โดยให้ครอบคลุม 100%

- Decision coverage
- Condition coverage
- Condition/decision coverage
- Modified Condition/Decision Coverage (MC/DC)

โดยส่งดังนี้

- ไฟล์ pdf แสดงตารางการออกแบบ test case คล้ายกับในสไลด์

#### ข้อ 2 : ทดสอบด้วยเทคนิค Decision และ Condition Coverage ต่าง ๆ

พิจารณาโปรแกรมการลงทะเบียน ซึ่งผู้ใช้งานกรอกข้อมูล อีเมล รหัสผ่าน และอายุ โดยโปรแกรมจะตรวจสอบความถูกต้องของข้อมูลก่อน ผู้ใช้ถึงจะลงทะเบียนได้ โดยจะตรวจสอบว่า อีเมลต้องมีตัวอักษร @ รหัสผ่านต้องมีความยาวอย่างน้อย 8 ตัวอักษร และผู้ใช้งานมีอายุอย่างน้อย 18 ปี

```
public boolean register(String email, String password, int age) {  
    if ( email.indexOf("@") >= 1 && password.length() >= 8 && age >= 18 ) {  
        // .... โค้ดลงทะเบียนสำเร็จ ...  
        return true;  
    }  
    return false;  
}  
  
def register(email, password, age):  
    if email.index("@") >= 1 and len(password) >= 8 and age >= 18:  
        # .... โค้ดลงทะเบียนสำเร็จ ...  
        return True  
    return False
```

จงออกแบบกรณีทดสอบด้วยเทคนิค structure-based ต่อไปนี้ โดยให้ครอบคลุม 100%

- i. Decision coverage
- ii. Condition coverage
- iii. Condition/decision coverage
- iv. Modified Condition/Decision Coverage (MC/DC)

โดยส่งดังนี้

- a) ไฟล์ pdf แสดงตารางการออกแบบ test case คล้ายกับในสไลด์
- b) ส่งผลการทดสอบด้วย unit testing framework ที่นิสิตถนัด ตาม test case ที่ได้ออกแบบไว้

### ข้อ 3 : ทดสอบเมธอดเช็คประเภทของสามเหลี่ยม ด้วยเทคนิค Condition Coverage

ให้ทดสอบเมธอด `getTriangleType()` ในภาษาจาวา หรือ `get_triangle_type()` ในภาษาไพธอน ด้วยเทคนิค white-box แบบ **condition coverage** และส่งดังนี้

- a) ส่งผลการทดสอบด้วย unit testing framework ที่นิสิตถนัด เพื่อทำ unit testing ด้วยเทคนิค condition coverage โดยให้ครอบคลุมเป็น 100%
- b) ทำตารางเปรียบเทียบจำนวน test case ที่ได้ออกแบบมา ระหว่างเทคนิคต่อไปนี้ (1) equivalence partitioning (2) statement coverage (3) decision coverage (4) condition coverage และอธิบายเหตุผลถึงความแตกต่าง

### ข้อ 4 : ทดสอบด้วยเทคนิค Loop, Path Coverage ต่าง ๆ

พิจารณาโปรแกรม `findFirst / find_first` ที่รับอะเรย์ (ในไพธอนคือ list) และจำนวนเต็ม `x` เพื่อหาว่ามี `x` ในอะเรย์หรือไม่ ถ้ามี จะคืนค่าเป็น index ของอะเรย์ที่เก็บ `x` ถ้าไม่มี จะคืนค่าเป็น -1

```
public int findFirst(int[] array, int x) {
    for (int i = 0; i < array.length; i++) {
        if (array[i] == x)
            return i;
    }
    return -1;
}

def find_first(array, x):
    for count, value in enumerate(array):
        if value == x:
            return count
    return -1
```

จงออกแบบกรณีทดสอบด้วยเทคนิค structure-based ต่อไปนี้ โดยให้ครอบคลุม 100%

- i. Loop coverage
- ii. Edge-pair coverage
- iii. McCabe Cyclomatic coverage

โดยส่งดังนี้

- a) ไฟล์ pdf แสดง control-flow graph และตารางการออกแบบ test case คล้ายกับในสไลด์
- b) ส่งผลการทดสอบด้วย unit testing framework ที่นิสิตถนัด ตาม test case ที่ได้ออกแบบไว้