

แบบฝึกปฏิบัติการครั้งที่ 4

1. จงศึกษาการใช้ class **Math** แล้วสร้างคลาสชื่อ **Q1** และเขียนโปรแกรม (method **main**) ที่รับพิกัด (x,y) ของจุดข้อมูล 4 จุด แล้วคำนวณหา
 - จุด centroid ของจุดข้อมูล 4 จุดนี้ (คำนวณจากค่าเฉลี่ยของค่า x ของจุดข้อมูลทั้ง 4 จุด และค่าเฉลี่ยของค่า y ของจุดข้อมูลทั้ง 4 จุด)
 - ผลรวมของระยะห่างจากจุด centroid ถึงจุดข้อมูลแต่ละจุด
 - ระยะห่างจากจุด centroid ถึงจุดข้อมูลที่น้อยสุดและมากที่สุด

ตัวอย่างผลลัพธ์การรัน

```
First point : 23.4 -9.47 ↵
Second point: -36.14 5 ↵
Third point : 0 3.14 ↵
Fourth point: -36.4 0↵
The centroid is ( -12.285, -0.33250000000000013 ).
Sum of distance is 98.16367890087943.
Shortest distance is 12.766341733245277.
Longest distance is 36.836301812885615.
```

2. จงศึกษาการใช้ class **String** แล้วสร้างคลาสชื่อ **Q2** และเขียนโปรแกรม (method **main**) ที่รับชื่อนามสกุลของผู้ใช้ และ อายุที่เป็นจำนวนเต็ม แล้วพิมพ์นามสกุล ชื่อ และอายุ ดังแสดงในตัวอย่าง (อาจใช้ **indexOf**, **substring**, **trim**)

ตัวอย่างผลลัพธ์การรัน

```
Enter firstname lastname age: Tom Smith 19 ↵
Smith, Tom is 19 years old.
```

3. จงสร้างคลาสชื่อ **Point** ที่เก็บ ของจุดใน 2 มิติเป็น instance variable ที่ใช้ได้ในคลาสนี้เท่านั้น และสร้าง method ต่อไปนี้
 - constructor ที่รับพารามิเตอร์เป็นพิกัด x, y

- method **distance** ที่รับพารามิเตอร์เป็นจุดอีกจุด และคืนค่าเป็นระยะห่างระหว่างจุด 2 จุดที่เป็น implicit parameter และ explicit parameter (คำนวณระยะทางแบบ Euclidean)
- method **translate** ที่รับพารามิเตอร์เป็นระยะตามแกน x และระยะตามแกน y ที่ให้เลื่อนจุด แล้วให้แก้พิกัดของจุดที่เป็น implicit parameter โดยเลื่อนจุดไปตามค่าพารามิเตอร์ที่กำหนด และคืนค่า **void**
- method **equals** ที่รับพารามิเตอร์เป็นออบเจกต์ในคลาส **Point** และคืนค่าเป็น Boolean ที่บอกว่า จุดทั้งสองอยู่ที่ตำแหน่งเดียวกันหรือไม่
- method **toString** ที่ไม่รับพารามิเตอร์และคืนค่าเป็นสตริงที่บอกพิกัดของจุด เช่น (3.1,-2.7)

ให้นิสิตกำหนด access specifier ของ variable และ method ให้เหมาะสม

จากนั้น ใช้คลาส Q3 (ที่มาจากชื่อของ Project) เป็น test class ซึ่ง method **main** ในคลาสนี้

- รับพิกัดของจุด 2 จุดมา แล้วสร้างออบเจกต์ในคลาส **Point** สำหรับ 2 จุดนั้น
- พิมพ์พิกัดของสองจุด และ ระยะห่างระหว่าง 2 จุด
- จากนั้น รับระยะตามแกน x และระยะตามแกน y แล้วเลื่อนจุดทั้งสองตามระยะนั้น
- แล้วพิมพ์พิกัดของสองจุด และ ระยะห่างระหว่าง 2 จุด พร้อมทั้งใช้ method **equals** บอกว่าเป็นจุดที่ตำแหน่งเดียวกันหรือไม่

ตัวอย่างผลลัพธ์การรัน

```
Point A: 3 -1.9↵
Point B: 0 2.1 ↵
A = (3.0,-1.9) B = (0.0,2.1) Distance = 5.0
Move the points: 0.5 -2.1↵
A = (3.5,-4.0) B = (0.5,0.0) Distance = 5.0
A and B are not at the same position.
```