



Task Lab05 ข้อ 1 (Lab05_1)

แบบฝึกปฏิบัติการครั้งที่ 5

Encapsulation and Constructor

จุดประสงค์

เมื่อผ่านปฏิบัติการนี้แล้ว นักศึกษาจะสามารถ

1. เข้าใจคลาส (Class) และอ็อบเจกต์ (Object)
2. เข้าใจการเข้าถึงข้อมูลและเมทอดสมาชิกของคลาส
3. เข้าใจการใช้งานของ Get Method, Set Method, Constructor Method , Instance Method และ Class Method

การส่งงาน

เข้าสู่เว็บ grader.cs.science.cmu.ac.th และ login ด้วย user และ password ที่แจกให้ทาง email

- เลือกเมนู Course > 66-204114 > เลือกข้อหรือ Task ที่ต้องการส่งงาน
- Upload ไฟล์ .java ที่มีชื่อเดียวกันกับชื่อ Task เช่น Lab05_1.java
- ให้เขียน comment เป็นรหัสนักศึกษาและชื่อไว้ด้านบนไฟล์

คำสั่ง

จงเขียนโปรแกรมเชิงวัตถุที่มีการใช้เมทอด Constructor เพื่อแก้ปัญหาต่อไปนี้

จงเขียนโปรแกรมเพื่อคำนวณปริมาตรของกระบอกน้ำทรงกระบอก 2 กระบอก กำหนดให้มีการเก็บข้อมูลของกระบอกน้ำแต่ละกระบอก ประกอบด้วย ชื่อของกระบอกน้ำ รัศมี ความสูงของกระบอก (ให้แยกเก็บเป็นเลขจำนวนจริง 2 ค่า) โดยการคำนวณหาปริมาตรของกระบอกน้ำคำนวณจาก $Volume = \pi r^2 h$

กำหนดให้

- กระบอกน้ำแรก ทราบข้อมูลอยู่แล้วคือ Size2 2 6
(ให้สร้าง constructor ที่ 1 คือ constructor แบบรับค่า 3 parameters)
- กระบอกน้ำที่สอง ยังไม่ทราบข้อมูล (ให้สร้าง constructor ที่ 2 คือ constructor แบบไม่รับค่าพารามิเตอร์)
- ให้สร้างเมทอดชื่อ setData() เพื่อรับค่าข้อมูลนำเข้าให้กับกระบอกน้ำที่สอง
- ให้สร้าง เมทอดชื่อ calVolume() เพื่อคำนวณหาปริมาตรของกระบอกน้ำทรงกระบอก
- แสดงผลปริมาตรของกระบอกน้ำเป็นตัวเลขทศนิยม 1 ตำแหน่ง
- ให้ใช้ค่า π ที่เป็น static constant ของคลาส Math ในภาษาจาวา
- ให้ใช้เมทอด pow() ที่เป็น static method (Class Method) ของคลาส Math ในภาษาจาวา

Input มี 2 บรรทัด

บรรทัดแรก เป็นสตริงขนาดไม่เกิน 20 อักขระ แทนชื่อกระบอกน้ำที่สอง

บรรทัดที่สอง เป็นเลขจำนวนจริง 2 ค่า แต่ละค่า คั่นด้วยช่องว่าง

- Radius เลขจำนวนจริง แทนจำนวนรัศมี, $1 \leq Radius \leq 100$
- Height เลขจำนวนจริง แทนจำนวนความสูงของกระบอกน้ำ, $1 \leq Height \leq 100$

Output มี 4 บรรทัด

บรรทัดแรก แสดงข้อความเมื่อสร้างวัตถุของกระบอกน้ำแรก "1st constructor"

บรรทัดสอง แสดงข้อความชื่อของกระบอกน้ำ และปริมาตร(Volume) ของกระบอกน้ำแรก เป็นเลขจำนวนจริง

บรรทัดสาม แสดงข้อความเมื่อสร้างวัตถุของกระบอกน้ำที่สอง "2nd constructor"

บรรทัดสี่ แสดงข้อความชื่อของกระบอกน้ำ และปริมาตร(Volume) ของกระบอกน้ำที่สอง เป็นเลขจำนวนจริง

ตัวอย่าง Input และ Output

ตัวอย่างที่	Input	Output
1	Size01 1.5 4	1st constructor Size02 75.4 2nd constructor Size01 28.3
2	Size05 5 15	1st constructor Size02 75.4 2nd constructor Size05 1178.1
3	Size10 10 4.4	1st constructor Size02 75.4 2nd constructor Size10 1382.3

Hint: การส่งงานบน เกรดเดอร์

- 1) การรับค่าเลขจำนวนจริง ให้ใช้ instance ของคลาส Scanner รับสตริง ด้วยคำสั่ง `.next()` แล้วค่อยใช้ static method (Class Method) ของคลาส Float หรือ Double ในการแปลงสตริงเป็นเลขจำนวนจริง

ดังตัวอย่างการรับเลขจำนวนจริงแบบ Double

```
Double.valueOf(input.next())
```

หรือ `Double.parseDouble(input.next())`

- 2) การแสดงผลเลขจำนวนจริงให้ใช้คำสั่งของคลาส `DecimalFormat` เพื่อแปลงเลขจำนวนจริงให้อยู่ในรูปสตริงตามรูปแบบ (pattern) ที่กำหนด ดังตัวอย่าง

```
import java.text.DecimalFormat;
```

```
DecimalFormat df = new DecimalFormat("0.0"); //กรณีแสดงผลด้วยสตริงด้วยรูปแบบเลขทศนิยม 0 ตำแหน่ง
```

```
System.out.println( df.format (ค่าหรือตัวแปรชนิดจำนวนจริง));
```



Task Lab05 ข้อ 2 (Lab05_2)

แบบฝึกปฏิบัติการครั้งที่ 5

Encapsulation and Constructor

จุดประสงค์

เมื่อผ่านปฏิบัติการนี้แล้ว นักศึกษาจะสามารถ

- 1) เข้าใจคลาส (Class) และอ็อบเจ็ค (Object)
- 2) เข้าใจการเข้าถึงข้อมูลและเมทอดสมาชิกของคลาส
- 3) เข้าใจการใช้งานของ Get Method, Set Method, Constructor Method , Instance Method และ Class Method

การส่งงาน

เข้าสู่เว็บ grader.cs.science.cmu.ac.th และ login ด้วย user และ password ที่แจกให้ทาง email

- เลือกเมนู Course > 66-204114 > เลือกข้อหรือ Task ที่ต้องการส่งงาน
- Upload ไฟล์ .java ที่มีชื่อเดียวกันกับชื่อ Task เช่น Lab05_2.java
- ให้เขียน comment เป็นรหัสนักศึกษาและชื่อไว้ด้านบนไฟล์

คำสั่ง

จงเขียนโปรแกรมเชิงวัตถุ เพื่อแก้ปัญหาต่อไปนี้ (ออกแบบโปรแกรมโดยให้พิจารณาว่าเมทอดใดควรสร้างเป็น เมทอดภายในคลาส และเมทอดใดควรสร้างเป็นเมทอดภายนอกคลาส) โดย

- 1) รับข้อมูลของคน 2 คน โดยรับ ชื่อ ส่วนสูง (int) น้ำหนัก (int) และเพศ (char โดย F แทนหญิง และ M แทนชาย)
- 2) หาค่าคนไหนที่มีน้ำหนักใกล้เคียงกับน้ำหนักมาตรฐานที่สุด แต่หากใกล้เคียงเท่ากันให้แสดงข้อความว่าเท่ากัน โดยน้ำหนักมาตรฐานจะพิจารณาตามเพศดังนี้
 - เพศชาย น้ำหนักมาตรฐาน = ส่วนสูง - 100
 - เพศหญิง น้ำหนักมาตรฐาน = ส่วนสูง - 110
- 3) พิมพ์ชื่อคนที่มีน้ำหนัก มีความใกล้เคียงกับน้ำหนักมาตรฐาน มากที่สุด แต่หากน้ำหนักของคนทั้งสอง มีความใกล้เคียงกับน้ำหนักมาตรฐานเท่ากัน ให้แสดงข้อความ "both"

Input มี 4 บรรทัด

บรรทัดแรก เป็นชื่อของคนแรก

บรรทัดที่สอง เป็นเพศ (char) ส่วนสูง (int) และ น้ำหนัก (int) ของคนแรก

บรรทัดที่สาม เป็นชื่อของคนที่สอง

บรรทัดที่สี่ เป็นเพศ (char) ส่วนสูง (int) และ น้ำหนัก (int) ของคนที่สอง

Output มี 1 บรรทัด

ชื่อคนที่มีน้ำหนัก มีความใกล้เคียงกับน้ำหนักมาตรฐาน มากที่สุด หรือข้อความ "both" ในกรณีน้ำหนักของคนทั้งสอง มีความใกล้เคียงกับน้ำหนักมาตรฐานเท่ากัน

ตัวอย่าง Input และ Output

ตัวอย่างที่	Input	Output	คำอธิบาย Output ที่ได้
1	Matinee F 160 60 Wattana M 170 60	both	คนแรก คือ Matinee สูง 160 น้ำหนัก 60 เพศ F (หญิง) คนที่สอง คือ Wattana สูง 170 น้ำหนัก 60 เพศ M (ชาย) คนแรก มีน้ำหนักมาตรฐาน คือ 50 มีน้ำหนักจริงคือ 60 ซึ่งต่างกันอยู่ 10 คนที่สอง มีน้ำหนักมาตรฐาน คือ 70 มีน้ำหนักจริงคือ 60 ซึ่งต่างกันอยู่ 10 ดังนั้นทั้งสองคนนี้มีน้ำหนักใกล้เคียงกับน้ำหนักมาตรฐาน ที่เท่ากัน
2	Matinee F 160 60 Kanokwan F 150 55	Matinee	คนแรก คือ Matinee สูง 160 น้ำหนัก 60 เพศ F (หญิง) คนที่สอง คือ Kanokwan สูง 150 น้ำหนัก 55 เพศ F (หญิง) คนแรก มีน้ำหนักมาตรฐาน คือ 50 มีน้ำหนักจริงคือ 60 ซึ่งต่างกันอยู่ 10 คนที่สอง มีน้ำหนักมาตรฐาน คือ 40 มีน้ำหนักจริงคือ 55 ซึ่งต่างกันอยู่ 15 ดังนั้น Matinee มีน้ำหนักใกล้เคียงกับน้ำหนักมาตรฐาน กว่า Kanokwan

```

public class Lab05_2 {
    public static void compareWeightbase(Person p1, Person p2){ //เมื่อกดภายนอกคลาส Person
        int diff1 = Math.abs(p1.getWeight() - p1.getStandardWeight());
        int diff2 = Math.abs(p2.getWeight() - p2.getStandardWeight());
        if (diff1 == diff2)
            System.out.println("both");
        else
            if (diff1 < diff2)
                System.out.println(p1.getName());
            else
                System.out.println(p2.getName());
    } // end method compareWeightbase()

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        String s = input.next().trim();
        Person p1 = new Person(s);           // create an object p1

        char sex = input.next().charAt(0);
        int H=input.nextInt();
        int W=input.nextInt();

        p1.setInfo(sex, H, W);                //เรียกใช้เมื่อกดภายในคลาส Person
    }
}

```

```

s = input.next().trim();
sex = input.next().charAt(0);

Person p2 = new Person(s, sex, input.nextInt(), input.nextInt()); // create an object p2
compareWeightbase(p1, p2); // เรียกใช้เมธอดภายนอกของคลาส Person

} // end method main()
} //end class Lab05_2

// ข้อนี้ให้เขียนเฉพาะส่วนของ Class
class Person{

} //end class Person

```



Task Lab05 ข้อ 3 (Lab05_3)

แบบฝึกปฏิบัติการครั้งที่ 5

Encapsulation and Constructor

จุดประสงค์

เมื่อผ่านปฏิบัติการนี้แล้ว นักเรียนจะสามารถ

1. เข้าใจคลาส (Class) และอ็อบเจกต์ (Object)
2. เข้าใจการเข้าถึงข้อมูลและเมทอดสมาชิกของคลาส
3. เข้าใจการใช้งานของ Get Method, Set Method, Constructor Method , Instance Method และ Class Method
4. เข้าใจการใช้งานเมทอดสมาชิกของคลาสประเภท Private

การส่งงาน

เข้าสู่เว็บ grader.cs.science.cmu.ac.th และ login ด้วย user และ password ที่แจกให้ทาง email

- เลือกเมนู Course > 65-204114 > เลือกข้อหรือ Task ที่ต้องการส่งงาน
- Upload ไฟล์ .java ที่มีชื่อเดียวกันกับชื่อ Task เช่น Lab05_3.java
- ให้เขียน comment เป็นรหัสนักศึกษาและชื่อไว้ด้านบนไฟล์

คำสั่ง

จงเขียนโปรแกรมเพื่อสร้างคลาส Stack โดยมีรายละเอียดดังนี้

5	10	7		
---	----	---	--	--



top

← ตัวอย่างกรณีมีข้อมูลในสแตก 3 ตัว ค่า top จะเท่ากับ 2

- เป็นคลาสสำหรับสร้างและจัดการกับ stack
- มี member data คือ
 - maxSize เป็นเลขจำนวนเต็ม เก็บค่าจำนวนสมาชิกที่สแตกสามารถเก็บได้
 - item[] เก็บข้อมูลสมาชิกของ stack ซึ่งจะเป็นเลขจำนวนเต็มไม่เกิน maxSize จำนวน
 - top เก็บข้อมูล index ของสมาชิกตัวสุดท้าย
- มี member method คือ
 - Stack(maxSize) เป็น constructor method ที่จะกำหนดให้ค่า top เป็น -1 โดยเป็นค่าในตอนเริ่มต้น เพื่อแสดงว่า stack ว่าง และกำหนดค่า maxSize เพื่อระบุจำนวนสมาชิกที่สแตกสามารถเก็บได้
 - void push(int x)
 - เป็นการเพิ่มข้อมูล x ลงใน stack ในตำแหน่ง top+1 และเพิ่มค่า top
 - ทั้งนี้การ push จะทำได้เมื่อ stack ไม่เต็ม (ตรวจสอบจากการเรียกใช้ isFull())
 - กรณีที่ stack เต็มให้แสดงข้อความว่า “full”
 - void pop()
 - เป็นการนำข้อมูล 1 จำนวนออกจาก stack ในตำแหน่ง top และลดค่า top ลง 1
 - พิมพ์ข้อมูลที่ pop ออกมาได้ เช่น pop 7
 - ทั้งนี้การ pop จะทำได้เมื่อ stack ไม่เป็น stack ว่าง (ตรวจสอบจากการเรียกใช้ isEmpty())

- กรณี stack ว่างให้แสดงข้อความว่า “empty”
 - void getTop() เพื่อ กรณีไม่มีข้อมูลให้แสดงข้อความ empty
 - เป็นการพิมพ์ข้อมูลที่อยู่ตำแหน่ง top ของ stack 1 จำนวน
 - ทั้งนี้ getTop() จะทำได้เมื่อ stack ไม่เป็น stack ว่าง (ตรวจสอบจากการเรียกใช้ isEmpty())
 - กรณี stack ว่างให้แสดงข้อความว่า “empty”
 - int getSize() ให้คืนค่าจำนวนข้อมูลในสแตก
 - boolean isEmpty() เป็น **private method** ของคลาสสแตกสำหรับการตรวจสอบว่า stack ว่างหรือไม่ (stack จะว่างเมื่อมีสมาชิกอยู่ใน stack 0 ตัว)

หากไม่มีข้อมูล (สแตกว่าง) ให้คืนค่า true หากไม่ว่างให้คืนค่า false
 - boolean isFull() เป็น **private method** ของคลาสสแตก สำหรับตรวจสอบว่า stack เต็มหรือไม่ (stack จะเต็มเมื่อมีสมาชิกอยู่ใน stack เท่ากับ maxSize ตัว)

หากสแตกเต็มให้คืนค่า true หากไม่เต็มให้คืนค่า false
 - void show() เป็นการพิมพ์ข้อมูลทั้งหมดใน stack ทีละค่า โดยคั่นด้วยช่องว่าง 1 ช่อง โดยไม่มีการนำข้อมูลออกจาก stack แต่หาก stack ว่าง ให้แสดงข้อความว่า “empty” ดังนี้
- ```
public void show(){
 if (isEmpty()){
 for(int i=0;i<=top;i++)
 System.out.print(item[i]+ " ");
 System.out.println();
 }
 else
 System.out.println("empty");
}
```

จึงเขียนโปรแกรมเชิงวัตถุเพื่อรับเมนูคำสั่ง และทำงานตามคำสั่ง จนกว่าผู้ใช้จะเลือกคำสั่งเป็น 6 จึงจะจบโปรแกรม

### ข้อมูลเข้า

บรรทัดแรก เป็นเลขจำนวนเต็ม N แทนขนาดของสแตกที่จะสร้าง โดย  $1 \leq N \leq 10000$

บรรทัดถัดไป จะเป็นเลขจำนวนเต็ม 1-2 ตัวต่อหนึ่งบรรทัดซึ่งเป็นการเรียกใช้งาน operation ต่างๆ และเรียกจบการทำงาน โดยมี 6 เป็นข้อมูลสุดท้ายเสมอ มีหน้าที่การทำงานดังนี้

| Operation | หน้าที่                                                                                                         |
|-----------|-----------------------------------------------------------------------------------------------------------------|
| 1 x       | เรียก push(x) ของ class stack เพื่อทำการเพิ่มข้อมูล x เข้าไปในสแตก<br>กรณีสแตกเต็มให้แสดงข้อความ full           |
| 2         | เรียก pop() เพื่อลบข้อมูลที่ตำแหน่งบนสุดออกจาก Stack และพิมพ์ข้อมูลที่ลบนั้น<br>กรณีไม่มีข้อมูลให้แสดงว่า empty |
| 3         | เรียก getTop() เพื่อพิมพ์ข้อมูลที่อยู่ตำแหน่งบนสุดออกจาก Stack กรณีไม่มีข้อมูลให้แสดงข้อความ empty              |
| 4         | เรียก getSize() เพื่อพิมพ์จำนวนข้อมูลในสแตก                                                                     |
| 5         | เรียก show() เพื่อพิมพ์ข้อมูลทั้งหมดในสแตก                                                                      |
| 6         | จบการทำงานของโปรแกรม                                                                                            |

### ข้อมูลออก

แต่ละบรรทัดเป็นการแสดงผลตามการดำเนินการ (Operation) ที่ได้รับ

ตัวอย่าง Input และ Output

| ตัวอย่างที่ | Input                                                                                    | Output                                                             |
|-------------|------------------------------------------------------------------------------------------|--------------------------------------------------------------------|
| 1           | 5<br>5<br>1 1<br>1 2<br>1 3<br>5<br>1 4<br>1 5<br>5<br>6                                 | empty<br>1 2 3<br>1 2 3 4 5                                        |
| 2           | 5<br>1 1<br>1 2<br>1 3<br>5<br>1 4<br>1 5<br>1 3<br>5<br>4<br>6                          | 1 2 3<br>full<br>1 2 3 4 5<br>5                                    |
| 3           | 5<br>1 1<br>1 2<br>1 3<br>5<br>3<br>2<br>1 4<br>1 5<br>1 3<br>5<br>2<br>2<br>2<br>5<br>6 | 1 2 3<br>3<br>pop 3<br>1 2 4 5 3<br>pop 3<br>pop 5<br>pop 4<br>1 2 |
| 4           | 10<br>1 1<br>1 2<br>5<br>2<br>3<br>2<br>2<br>6<br>6                                      | 1 2<br>pop 2<br>1<br>pop 1<br>empty                                |





Task Lab05 ข้อ 4 (Lab05\_4)

แบบฝึกปฏิบัติการครั้งที่ 5

Encapsulation and Constructor

จุดประสงค์

เมื่อผ่านปฏิบัติการนี้แล้ว นักศึกษาจะสามารถ

1. เข้าใจคลาส (Class) และอ็อบเจ็ค (Object)
2. เข้าใจการเข้าถึงข้อมูลและเมทอดสมาชิกของคลาส
3. เข้าใจการใช้งานของ Get Method, Set Method, Constructor Method , Instance Method และ Class Method
4. เข้าใจการใช้งานเมทอดสมาชิกของคลาสประเภท Private

การส่งงาน

เข้าสู่เว็บ grader.cs.science.cmu.ac.th และ login ด้วย user และ password ที่แจกให้ทาง email

- เลือกเมนู Course > 66-204114 > เลือกข้อหรือ Task ที่ต้องการส่งงาน
- Upload ไฟล์ .java ที่มีชื่อเดียวกันกับชื่อ Task เช่น Lab05\_4.java
- ให้เขียน comment เป็นรหัสนักศึกษาและชื่อไว้ด้านบนไฟล์

คำสั่ง

จงเขียนโปรแกรมเชิงวัตถุเพื่อรับจำนวนวิชา (N) ที่นักศึกษาคนหนึ่งลงทะเบียนจาก main จากนั้นให้รับข้อมูลเกรด และหน่วยกิต ของแต่ละวิชา จนครบ N วิชา เพื่อแสดงผลลัพธ์คือค่า GPA ของนักศึกษานั้น  
กำหนดให้ออกแบบคลาส Course ที่มีการใช้

- เมทอดประเภท set และ get ในการกำหนดและเรียกใช้ค่าเกรดและหน่วยกิต
- เมทอดชนิด Private ในการกำหนดค่าแต้มประจำเกรด ของแต่ละเกรดตามเงื่อนไขดังนี้

| เกรด | แต้มประจำเกรด |
|------|---------------|
| A    | 4             |
| B    | 3             |
| C    | 2             |
| D    | 1             |
| F    | 0             |

ตัวอย่าง Input และ Output และการคำนวณค่า GPA

ตัวอย่างข้อมูลนำเข้า

3  
A 3  
B 4  
F 3

### ตัวอย่างการคิดคำนวณค่า GPA

|           | เกรด(Grade) | แต้มประจำเกรด (GradePoint) | หน่วยกิต(Credit) | แต้มที่ได้ (TotalPoint) |
|-----------|-------------|----------------------------|------------------|-------------------------|
| Subject 1 | A           | 4                          | 3                | 12                      |
| Subject 2 | B           | 3                          | 4                | 12                      |
| Subject 3 | F           | 0                          | 3                | 0                       |
| รวม       |             |                            | 10               | 24                      |

$$GPA = \text{แต้มที่ได้รวม/หน่วยกิตรวม} = 24/10 = 2.40$$

### ตัวอย่างข้อมูลส่งออก

A 4 3 12  
B 3 4 12  
F 0 3 0  
10 24 2.40

Input มี N+1 บรรทัด

บรรทัดแรก เป็นเลขจำนวนเต็ม N แทนจำนวนวิชา  $1 \leq N \leq 1000$

บรรทัดที่ 2 ...N+1 บรรทัด แต่ละบรรทัดประกอบด้วยค่าเกรด(Grade) และหน่วยกิตของแต่ละวิชา (Cr)

โดย  $Grade \in \{'A', 'B', 'C', 'D', 'F'\}$  และ  $1 \leq Cr \leq 20$

Output มี N+1 บรรทัด แต่ละค่าคั่นด้วยช่องว่าง

N บรรทัดแรกประกอบด้วยลำดับ Grade, GradePoint, Credit และ TotalPoint แต่ละวิชา

บรรทัดสุดท้าย ประกอบด้วย หน่วยกิตรวม แต้มที่ได้รวม และ GPA

### ตัวอย่าง Input และ Output

| ตัวอย่างที่ | Input                                                     | Output                                                                                              |
|-------------|-----------------------------------------------------------|-----------------------------------------------------------------------------------------------------|
| 1           | 3<br>A 3<br>B 4<br>F 3                                    | A 4 3 12<br>B 3 4 12<br>F 0 3 0<br>10 24 2.40                                                       |
| 2           | 8<br>C 3<br>C 4<br>D 4<br>A 3<br>C 3<br>B 3<br>C 3<br>B 3 | C 2 3 6<br>C 2 4 8<br>D 1 4 4<br>A 4 3 12<br>C 2 3 6<br>B 3 3 9<br>C 2 3 6<br>B 3 3 9<br>26 60 2.31 |

กำหนดให้ใช้

DecimalFormat df = new DecimalFormat("0.00"); //ในการแสดงผลค่าปริมาตรแบบทศนิยม 2 ตำแหน่ง