



มหาวิทยาลัยสงขลานครินทร์  
คณะวิศวกรรมศาสตร์

---

สอบปลายภาค: ภาคการศึกษาที่ 2

ปีการศึกษา: 2546

วันที่สอบ: 17 กุมภาพันธ์ 2547

เวลาสอบ: 13.30 – 16.30 น.

รหัสวิชา: 240-101

ห้องสอบ: R200, R201, R300

ชื่อวิชา: Introduction to Computer Programming

---

คำสั่ง: อ่านรายละเอียดของข้อสอบ และคำแนะนำให้เข้าใจก่อนเริ่มทำข้อสอบ

อนุญาต: เครื่องเขียนต่างๆ เช่น ปากกา หรือดินสอ

ไม่อนุญาต: หนังสือ, เอกสารใดๆ และเครื่องคิดเลข

เวลา: 3 ชั่วโมง (180 นาที)

#### คำแนะนำ

- ข้อสอบมี 14 หน้า (รวมใบปะหน้า) แบ่งเป็น 4 ตอน คะแนนรวม 100 คะแนน
- คำตอบทั้งหมดจะต้องเขียนลงในข้อสอบ รวมทั้ง เขียนชื่อและรหัสให้ชัดเจนทุกแผ่น ถ้าแผ่นใดของข้อสอบไม่มีชื่อและรหัส นักศึกษาจะถูกหักคะแนน 0.5 คะแนนต่อแผ่น
- คำตอบส่วนใดอ่านไม่ออก จะถือว่าคำตอบนั้นผิด
- การเขียนโปรแกรม ให้ใช้ภาษา C++
- อ่านคำสั่งในแต่ละข้อให้เข้าใจก่อนลงมือทำ
- หากข้อใดเขียนตอบไม่พอ ให้เขียนเพิ่มเติมที่ด้านหลังของหน้านั้นเท่านั้น

**ตอนที่ 1 ( 20 คะแนน : 35 นาที)**

1. จงบอกความหมายของการประกาศตัวแปรต่อไปนี้ (4 คะแนน)

1.1 int score[10];

---

---

1.2 float volume[50][60];

---

---

1.3 char username[100];

---

---

1.4 char student[10][30];

---

---

2. จงเขียนการประกาศต่อไปนี้ (6 คะแนน)

2.1 ตัวแปร name ใช้เก็บชื่อคนที่มีความยาวไม่เกิน 29 ตัวอักษร

---

---

2.2 ตัวแปร price เป็นอะเรย์ขนาด 5 อิลิเมนต์ของตัวเลขจำนวนเต็ม ให้มีค่าเริ่มต้นเป็น 1, 2, 3, 4 ,5 ตามลำดับ

---

---

2.3 ตัวแปร logger เก็บจำนวนเสื้อที่อยู่ในตู้เก็บของ ตู้มีจำนวน 360 ช่อง แบ่งเป็น 12 แถว 30 สดมภ์

---

---

2.4 ตัวแปร stock เก็บจำนวนสินค้าเป็นเลขจำนวนเต็ม มีสินค้าทั้งหมด 10 ชนิด และกำหนดให้ค่าเริ่มต้นจำนวนสินค้าทุกตัวมีค่าเป็น 100

---

---

2.5 ตัวแปร staff เก็บชื่อคนจำนวน 3 ชื่อ แต่ละชื่อมีความยาวสูงสุดไม่เกิน 40 ตัวอักษร ให้มีค่า เริ่มต้นคือ sombhat, somboon และ somchai ตามลำดับ

---

---

2.6 ตัวแปรชื่อ room เป็นอะเรย์ 2 มิติของตัวเลขจำนวนเต็ม ใช้เก็บผู้เข้าพักในห้องพักของโรงแรม ซึ่งมีห้องพักจำนวน 300 ห้อง แบ่งเป็น 10 ชั้น ชั้นละ 30 ห้อง ให้ค่าเริ่มต้นเป็นศูนย์ทั้งหมด

---

---

3. จากส่วนของโปรแกรมต่อไปนี้ จงเขียนผลลัพธ์ที่ปรากฏบนจอคอมพิวเตอร์ (3 คะแนน)

```
int i, counter[5] = { 1, 2, 3, 4, 5 };
for (i=0; i<4; ++i) {
    counter[i]= counter[i+1];
}
for (i=0; i<5; ++i) {
    cout << counter[i] * counter[i] << " ";
}
```

---

---

---

---

---

4. จากส่วนของโปรแกรมต่อไปนี้ จงเขียนผลลัพธ์ที่ปรากฏบนจอคอมพิวเตอร์ (3 คะแนน)

```
int i, j, sum=0;
int table1[2][3] = { { 1, 2, 3 }, {4, 5, 6} };
int table2[2][3] = { { 10, 20, 30 }, {40, 50, 60} };
for (i=0; i<2; ++i) {
    for (j=0; j<3; ++j) {
        table1[i][j]= table1[i][j] * table2[i][j];
    }
}
```

```
for (i=0; i<2; ++i) {
    for (j=0; j<3; ++j) {
        cout << table1[i][j] << " ";
        sum+=table1[i][j];
    }
    cout << endl;
}
cout << "sum = " << sum << "endl";
```

---

---

---

---

---



**ตอนที่ 2 (25 คะแนน: 35 นาที)**

ตารางต่อไปนี้เป็นสำหรับอาจารย์ผู้ตรวจเท่านั้น

ข้อ	คะแนนเต็ม	คะแนน
1	14	
2	6	
3	5	
<b>รวม</b>	<b>25</b>	

**คำสั่ง** ข้อสอบในตอนที่ 2 นี้มีทั้งหมด 3 ข้อ จงอ่านคำสั่งในแต่ละข้อให้เข้าใจก่อนลงมือทำ

1. จงหาค่าผลลัพธ์ที่แสดงออกทางหน้าจอคอมพิวเตอร์ของส่วนหนึ่งของโปรแกรมต่อไปนี้ (14 คะแนน)

ข้อ	ส่วนหนึ่งของโปรแกรม	ผลลัพธ์ที่แสดงออกทางหน้าจอ
1.1	<pre>int x = 5; int y = 2; int *xPtr = &amp;x; int *yPtr = &amp;y;  cout &lt;&lt; *xPtr &lt;&lt; endl;  *yPtr = 8 - y; cout &lt;&lt; y &lt;&lt; endl;  int z = x - 1; int *zPtr; zPtr = &amp;z; cout &lt;&lt; *zPtr &lt;&lt; endl;</pre>	<p>.....</p> <p>.....</p> <p>.....</p>
1.2	<pre>float f[4] = {7, -1, 2, 21}; float *fPtr = f;  cout &lt;&lt; (*fPtr)+2 &lt;&lt; endl;  cout &lt;&lt; fPtr[0] &lt;&lt; endl;  fPtr[2]++; cout &lt;&lt; f[2] &lt;&lt; endl;  fPtr = fPtr+2; cout &lt;&lt; fPtr[1] &lt;&lt; endl;</pre>	<p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p>

2. จงเติมคำสั่งในช่องว่างของโปรแกรมเพื่อให้ได้ผลลัพธ์ดังที่กำหนด (6 คะแนน)

ข้อ	โปรแกรม	ผลลัพธ์ที่แสดงออกทางหน้าจอ
2.1	<pre>#include &lt;iostream.h&gt; char* mys (char *sPtr){     char *dPtr = sPtr;     while(*dPtr != 'o' &amp;&amp; *dPtr != '\0')         _____;     return dPtr; } int main(){     char *des = "The league of";     char *src = "extraordinary";     des = mys(src) ;     cout &lt;&lt; des &lt;&lt; endl;     return 0; }</pre>	<b>ordinary</b>

2.2	<pre>#include &lt;iostream.h&gt; void calc (int *xPtr){     _____; }  int main () {     int x = 10;     calc(_____);     cout &lt;&lt; x &lt;&lt; endl;     return 0; }</pre>	6
-----	---	---

3. จงเขียนคำสั่งในภาษา C++ เพื่อให้ทำงานต่อไปนี้ (5 คะแนน)

- ประกาศพอยน์เตอร์ชนิดตัวอักษรชื่อ cPtr

.....

.....

- ประกาศพอยน์เตอร์ชื่อ dPtr ซึ่งเก็บค่าตำแหน่งของหน่วยความจำหนึ่งโดยหน่วยความจำนี้ถูกจองไว้เพื่อเก็บค่าจำนวนทศนิยม

.....

.....

- กำหนดให้พอยน์เตอร์ชนิดจำนวนเต็มชื่อ xPtr เก็บตำแหน่งของตัวแปรชนิดจำนวนเต็มชื่อ x โดยสมมุติว่ามีการประกาศพอยน์เตอร์ xPtr และ ตัวแปร x ก่อนหน้านี้แล้ว

.....

.....

- คำนวณหน่วยความจำแบบหลายบล็อกที่เก็บค่าของอะเรย์ชนิดจำนวนเต็มให้กับคอมพิวเตอร์ โดยค่าตำแหน่งแรกของหน่วยความจำนี้ถูกเก็บไว้ในพอยน์เตอร์ชื่อ arrPtr และสมมุติว่ามีการประกาศพอยน์เตอร์ arrPtr ก่อนหน้านี้แล้ว

.....

.....

- แสดงค่าตำแหน่งในหน่วยความจำของพอยน์เตอร์ชื่อ bPtr ออกทางหน้าจอคอมพิวเตอร์ โดยสมมุติว่ามีการประกาศพอยน์เตอร์ bPtr ก่อนหน้านี้แล้ว

.....

.....

**/\*\* จบตอนที่ 2 อย่าลืมเขียนชื่อ รหัส และ Section ลงในข้อสอบทุกแผ่น\*\*/**



**ตอนที่ 3 (20 คะแนน: 50 นาที)**

ในการคำนวณคะแนนครั้งหนึ่ง โปรแกรมเมอร์จะต้องเขียนโปรแกรมที่รับคะแนนของนักศึกษาทั้งหมด 10 กลุ่ม กลุ่มละ 10 คน และโปรแกรมจะต้องทำการเพิ่มคะแนนของกลุ่มที่มีคะแนนเฉลี่ยของกลุ่มน้อยกว่า 50 คะแนน โดยให้ทุกคนในกลุ่มมีคะแนนเพิ่มขึ้น 2 คะแนน

```
#include<iostream.h>

#define MEMBER_PER_GROUP _____ /*1.1*/ //จำนวนนักเรียนต่อกลุ่ม
#define MEMBER_TOTAL _____ /*1.2*/ //จำนวนนักเรียนทั้งหมด

int* find_group_average(int data[], int num, int member_per_group);
void upGroup(int data[], int member_per_group, int value);

int main(){

    int score[MEMBER_TOTAL];
    int *group_avg; //ตัวแปรไว้เก็บคะแนนเฉลี่ยของแต่ละกลุ่ม

    for(int i = 0; i < MEMBER_TOTAL; i++){
        //ทำการรับค่าคะแนนของแต่ละคน
        _____ /*2.1*/
    }

    _____ /*2.2*/ = find_group_average(_____ /*3*/);

    for(int i = 0; i < MEMBER_TOTAL/MEMBER_PER_GROUP; i ++){
        if(_____ /*4*/)
            //ทำการเพิ่มคะแนนให้กับนักเรียนทั้งกลุ่ม
            upGroup(&score[i*MEMBER_PER_GROUP], MEMBER_PER_GROUP, 2);
    }

    cout << "After adjusting...." << endl;

    for(int i = 0; i < MEMBER_TOTAL; i++){
        cout << score[i] << endl;
    }

    _____ /*5*/

    return 0;
}
```

ข้างล่าง แสดงส่วนของโปรแกรม ที่ได้ออกแบบฟังก์ชัน และมีการนิยามฟังก์ชันไว้บางส่วน โดยนอกจากฟังก์ชัน main() กำหนดให้มีฟังก์ชันร่วมทำงาน 2 ฟังก์ชันด้วยกัน คือ

- find\_group\_average() ทำการคำนวณคะแนนเฉลี่ยของแต่ละกลุ่ม โดยจะคืนค่ากลับเป็น address ของ array ผลลัพธ์ของการคำนวณ
  - data ข้อมูลที่ต้องการประมวลผล
  - num จำนวนข้อมูลที่ต้องการประมวล
  - member\_per\_group จำนวนของนักศึกษาต่อกลุ่ม
- upGroup() ทำการเพิ่มคะแนนของนักศึกษาจำนวนหนึ่งกลุ่ม
  - data ข้อมูลของสมาชิกในกลุ่มที่ต้องการให้เพิ่มค่า โดยให้ data[0] ถือเป็นสมาชิกคนแรก
  - member\_per\_group จำนวนของนักศึกษาในกลุ่ม
  - value คะแนนที่ต้องการเพิ่มให้กับทั้งกลุ่ม (ทุกคนได้รับการเพิ่มคะแนนเท่ากัน)

ข้อที่ 1 จงเขียน code ที่จะต้องเติมในช่องว่างที่กำหนดไว้ใน code ของโปรแกรม เพื่อให้โปรแกรมทำงานได้ตามต้องการ (รวม 10 คะแนน)

/\*1.1\*/ .....

/\*1.2\*/ .....

/\*2.1\*/ .....

/\*2.2\*/ .....

/\*3\*/ .....

/\*4\*/ .....

/\*5\*/ .....

ชื่อ \_\_\_\_\_ รหัสนักศึกษา \_\_\_\_\_ Section \_\_\_\_\_

ข้อที่ 2 จงเขียนนิยามของฟังก์ชัน find\_group\_average() อย่างเหมาะสม (5 คะแนน)

คำตอบ

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

ข้อที่ 2 จงเขียนนิยามของฟังก์ชัน upGroup() อย่างเหมาะสม (5 คะแนน)

คำตอบ

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

***/\*\* จบตอนที่ 3 อย่าลืมเขียนชื่อ รหัส และ Section ลงในข้อสอบทุกแผ่น\*\*/***

**ตอนที่ 4 (35 คะแนน: 1 ชั่วโมง)****ข้อที่ 1** จงนำตัวอักษรของคำตอบทางขวาที่เหมาะสมที่สุดหนึ่งคำตอบ มาเติมในช่องว่างทางซ้าย

(5 คะแนน: 10 นาที)

## ส่วนที่ 1

- |   |   |
|---|---|
| _____ 1. class Male : Human { };              | A. Abstraction, Encapsulation, Modularity |
| _____ 2. OOP                                  | และ Hierarchy                             |
| _____ 3. เครื่องบิน รถยนต์ ถือว่าเป็นยานพาหนะ | B. Hierarchy                              |
| _____ 4. การนำ class ที่เคยพัฒนาไว้ มาใช้กับ  | C. Constructor                            |
| ซอฟต์แวร์ตัวใหม่ เรียกว่า การ Reuse           | D. Modularity                             |
| _____ 5. หลักสำคัญของ OOP                     | E. Abstraction                            |
|   | F. แนวคิดการเขียนโปรแกรมเชิงวัตถุ         |

## ส่วนที่ 2

- |  |                                       |
|--|---------------------------------------|
| _____ 1. data member ชนิด private                      | A. class                              |
| _____ 2. ฟังก์ชันที่เรียกใช้อัตโนมัติเมื่อสร้าง Object | B. data member ชนิด static            |
| _____ 3. ตัวแปรที่ Object จาก class เดียวกัน ใช้       | C. ไม่สามารถเข้าถึงได้โดย object อื่น |
| ร่วมกันได้   | D. Constructor                        |
| _____ 4. องค์ประกอบของ Object                          | E. properties และ methods             |
| _____ 5. ต้นแบบในการสร้าง Object                       | F. Object และ ฟังก์ชัน                |

**ข้อที่ 2** จาก Code ที่กำหนดให้ในหน้าถัดไป (triangle.cpp) จงแก้โปรแกรมอย่างเหมาะสมที่สุด เพื่อให้สามารถนำไปคอมไพล์ และใช้งานได้ถูกต้อง

(20 คะแนน: 30 นาที)

นักศึกษาควรประยุกต์ใช้เนื้อหาดังต่อไปนี้

การใช้งาน public, private

การนิยามฟังก์ชัน

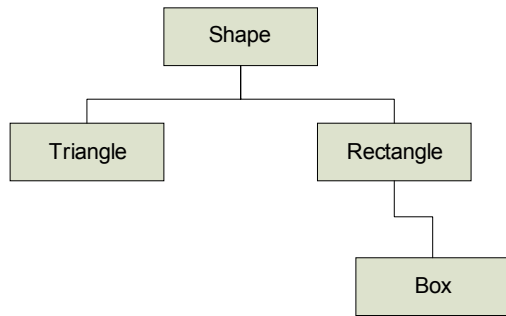
การใช้งาน Object

คำแนะนำ: โปรแกรมที่เขียนมาจะให้คะแนนตามส่วนของความถูกต้องและเหมาะสม ดังนั้นนักศึกษาไม่ควรเว้นว่าง



**ข้อที่ 3** ในการเขียนโปรแกรมเพื่อคำนวณหาพื้นที่ของรูปร่างสองมิติ และปริมาตรของรูปร่างสามมิติ

โปรแกรมเมอร์ต้องการสร้างโครงสร้างการสืบทอดของ Object ดังแสดงในแผนภาพข้างล่าง



กล่าวคือ Shape เป็นรูปร่างทั่วไป ไม่เจาะจง ในขณะที่สามารถนำไปสร้างต่อเป็น สามเหลี่ยม (Triangle) หรือ สี่เหลี่ยม (Rectangle) และ สี่เหลี่ยมสามารถนำไปสร้างต่อเป็นกล่อง (Box)

ให้นักศึกษาอธิบาย **พอสังเขป** ว่า Object แต่ละตัวควรมี เมธอด (Method) อะไรบ้าง โดยให้ยกตัวอย่าง code ในการสืบทอด ประกอบคำอธิบาย

(10 คะแนน: 20 นาที)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

*/\*\* จบตอนที่ 4 อย่าลืมเขียนชื่อ รหัส และ Section ลงในข้อสอบทุกแผ่น\*\*/*