

PRINCE OF SONGKLA UNIVERSITY  
FACULTY OF ENGINEERING



Midterm Examination: ภาคการศึกษาที่ 1

Academic Year: 2547

Date: 8 สิงหาคม พ.ศ. 2547

Time: 13.30 – 16.30 น.

Subject Number: 240-204

Room: R201, R300

Subject Title: Data Structures and Computer Programming Techniques

ทฤษฎีในการสอบ มีโทษขั้นต่ำ คือ ปรับตกในรายวิชาที่ทฤษฎี และพักการเรียน 1 ภาคการศึกษา

อ่านรายละเอียดของข้อสอบ และคำแนะนำให้เข้าใจก่อนเริ่มทำข้อสอบ

รายละเอียดของข้อสอบ:

เวลา 3 ชั่วโมง (180 คะแนน: 180 นาที)

ข้อสอบมีทั้งหมด 9 หน้า (ไม่รวมหน้านี้) ประกอบด้วยคำถามจำนวน 4 ข้อ

สิ่งที่สามารถนำเข้าห้องสอบได้:

อนุญาต: เครื่องเขียนต่างๆ เช่น ปากกา หรือดินสอ

ไม่อนุญาต: หนังสือ และเครื่องคิดเลข

คำแนะนำ:

- พยายามทำทุกข้อ
- คำตอบทั้งหมดจะต้องเขียนลงในช่องว่างที่เว้นไว้ในข้อสอบ
- คำตอบส่วนใดอ่านไม่ออก จะถือว่าคำตอบนั้นผิด
- อ่านคำสั่งในแต่ละข้อให้ชัดเจนว่า เขียนโปรแกรมบางส่วน เขียนฟังก์ชัน หรือเขียนทั้งโปรแกรม รวมไปถึงข้อกำหนดเพิ่มเติม และหมายเหตุในข้อนั้นๆ
- การเขียนโปรแกรมในแต่ละข้อ อาจจะไม่ต้องเขียนตามคำสั่งย่อยทั้งหมด แต่คะแนนจะลดลงตามส่วน และหากในข้อใหญ่หนึ่งข้อ นักศึกษาไม่สามารถทำข้อย่อยข้อแรกๆ ได้ นักศึกษาสามารถทำข้อย่อยหลังๆ โดยให้อ้างอิงเหมือนนักศึกษาทำข้อย่อยข้อแรกๆ ได้
- การเขียน code จะต้องตั้งชื่อตัวแปรให้เหมาะสม และมี comment ในจุดสำคัญต่างๆ โดยให้ทั้งหมดเป็นไปตามหลักการเขียนโปรแกรมที่ดี

**ข้อที่ 1 ARRAY, POINTER, ENUMERATED TYPE****(30 คะแนน: 30 นาที)**

1.1 ให้อาเรย์ data เก็บจำนวนตัวเลขในช่วง 1...10 จงเขียนส่วนของโค้ด เพื่อคำนวณความถี่ของค่าตัวเลขต่างๆ ภายในอาเรย์ data โดยเก็บค่าผลลัพธ์ดังกล่าวไว้ในอาเรย์ frequency (10 คะแนน)

```
const int arraySize = 10;
int data[arraySize] = {1, 6, 2, 3, 2, 5, 4, 5, 6 ...};
int frequency[arraySize];
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

1.2 จากส่วนของโค้ดข้างล่าง จงเขียนส่วนของโค้ด เพื่อสร้างตัวแปรตามโครงสร้างข้อมูล SUIT ข้างล่าง แล้วให้ค่าเริ่มต้นเป็น Clubs (5 คะแนน)

```
typedef enum {Hearts, Diamonds, Clubs, Spades} SUIT;
```

.....

.....

.....

.....

.....

1.3 จากส่วนของโค้ดข้างล่าง จงเขียนส่วนของโค้ด ที่ใช้งาน pointer notation (\*) และ array notation ([]) เพื่ออ้างอิง ข้อมูลใน b เปรียบเทียบกัน (5 คะแนน)

```
int b[] = { 10, 20, 30, 40 };
int *bPtr = b; // set bPtr to point to array b

for ( int i = 0; i < 4; i++ )
    cout << "Array Notation: b[" << i << "] = " <<
    _____ << '\n';
for ( int offset1 = 0; offset1 < 4; offset1++ )
    cout << "Pointer Notation: b ( " << offset1 << ") = " <<
    _____ << '\n';
```

**ข้อที่ 2 FUNCTIONS, CALL BY REFERENCE, REFERENCE PARAMETER, STATIC VARIABLE**

**(30 คะแนน: 30 นาที)**

2.1 จงเขียนฟังก์ชัน `capital` ที่รับตัวอักษรภาษาอังกฤษ 1 ตัวเข้ามาแปลงให้เป็นอักษรตัวใหญ่ โดยส่งตัวอักษรผ่านฟังก์ชัน แบบ `Call by Reference` และ `Reference Parameter` พร้อมทั้งแสดงส่วนของโค้ดที่เรียกใช้ฟังก์ชันดังกล่าวด้วย (10 คะแนน)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

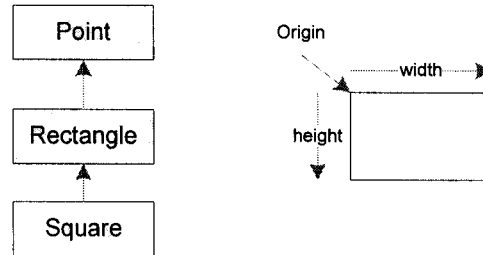
2.2 จงปรับฟังก์ชัน fibonacci ต่อไปนี้ ให้นับจำนวนครั้งที่มีการเรียกใช้ฟังก์ชันนี้ในโปรแกรม โดยใช้ประโยชน์จากตัวแปรแบบ static พร้อมทั้งแสดงส่วนของโค้ดที่เรียกใช้ฟังก์ชันดังกล่าวด้วย (10 คะแนน)

```
// recursive definition of function fibonacci
unsigned long fibonacci( unsigned long n )
{
// base case
  if ( n == 0 || n == 1 )
    return n;
// recursive step
  else
    return fibonacci( n - 1 ) + fibonacci( n - 2 );
} // end function fibonacci
```

**ข้อที่ 3 OOP: INHERITANCE****(40 คะแนน: 40 นาที)**

แผนภาพข้างล่าง แสดงการสืบทอดของ class ทั้งหมด 3 class กล่าวคือ Point, Rectangle (สี่เหลี่ยมด้านไม่เท่า) และ Square (สี่เหลี่ยมด้านเท่า)

จะเห็นว่า สี่เหลี่ยมหนึ่งรูปนั้น สามารถแสดงตัวเองได้ด้วยจุด Origin และความยาว ความสูง และจากข้อมูลที่มีอยู่ เราสามารถที่จะคำนวณหาพื้นที่ของรูปสี่เหลี่ยมนั้น ๆ ได้



จงเขียน Code อย่างสมบูรณ์สำหรับ class Rectangle (25 คะแนน) และ class Square (15 คะแนน) เพื่อใช้ประโยชน์จากการสืบทอด (Inheritance) ในภาษา C++ อย่างเต็มที่ และเหมาะสม

```
//-----Point.h-----
#ifndef POINT_H
#define POINT_H

class Point
{
public:
    Point( int x, int y );
    int getX();
    int getY();

private:
    int x, y;
};

#endif

//-----Point.cpp-----
#include "point.h"

Point::Point( int x, int y )
{
    this->x = x;
    this->y = y;
}

int Point::getX()
{
    return x;
}

int Point::getY()
{
    return y;
}
```

**ข้อที่ 4 OOP****(80 คะแนน: 80 นาที)**

จาก class Point ในข้อที่ 3 และ ฟังก์ชัน main() ที่กำหนดให้ข้างล่าง

```

#include <iostream>
#include <new>

#include "line.h"

using std::cout;

int main(){
    const int size = 3;
    Line *lines[size];
    lines[0] = new Line(Point(5, 5), Point(2, 4));
    lines[1] = new Line(Point(10, 6), Point(2, 2));
    lines[2] = new Line(Point(1, 6), Point(2, 2));

    for(int i = 0; i < size; i ++){
        cout << "Line ID:" << lines[i]->getId() << " has length = " <<
lines[i]->length() << "\n";

        cout << "The longest line is Line ID:" <<
(Line::getLongestLine(lines, size))->getId() << "\n";

        for(int i = 0; i < size; i ++){
            delete lines[i];
        }

        return 0;
    }
}

```

ผลการรันโปรแกรม ดังแสดงข้างล่าง

```

Line ID:1 has length = 3.16228
Line ID:2 has length = 8.94427
Line ID:3 has length = 4.12311
The longest line is Line ID:2

```

4.1 จงเขียน class Line ที่ทำให้โปรแกรมให้ผลการรันตามที่กำหนด โดยมีข้อกำหนดเพิ่มเติมให้ฟังก์ชัน getLongestLine() เรียกใช้ฟังก์ชัน getLongerLine() (10 คะแนน) ให้กับ class Line เพื่อคำนวณค่าเส้นตรงสองเส้นที่ส่งมาเป็น parameters เส้นใดยาวกว่านั้น โดยกำหนดให้ prototype เป็นดังนี้

```
Line* getLongerLine(Line* line1, Line* line2);
```

โดยมีอัตราส่วนการให้คะแนนดังนี้

การใช้งาน static data และ static method	15 คะแนน
การออกแบบ class ตามหลัก OOP	15 คะแนน
ฟังก์ชัน getLongerLine()	15 คะแนน
ฟังก์ชัน getLongestLine()	20 คะแนน

4.2 จงวิเคราะห์ว่า มีเหตุผลสนับสนุนใดบ้างที่จะกำหนดให้ `getLongerLine()` เป็น `private` และเป็น `static` (15 คะแนน)