



มหาวิทยาลัยราชภัฏนครินทร์
คณะวิศวกรรมศาสตร์

สอบปลายภาค: ภาคการศึกษาที่ 1

ปีการศึกษา: 2549

วันที่สอบ: 6 สิงหาคม 2549

เวลาสอบ: 13.30 – 16.30 น.

รหัสวิชา: 240-204

ห้องสอบ: หัวหุ่น, R300

ชื่อวิชา: Data Structures and Computer Programming Techniques

คำสั่ง: อ่านรายละเอียดของข้อสอบ และคำแนะนำให้เข้าใจก่อนเริ่มทำข้อสอบ

อนุญาต: เครื่องเขียนต่างๆ เช่น ปากกา หรือดินสอ เข้าห้องสอบ

ไม่อนุญาต: หนังสือ หรือเครื่องคิดเลขเข้าห้องสอบ และเอกสารใดๆ เข้าห้องสอบ

เวลา: 3 ชั่วโมง (180 นาที)

คำแนะนำ

- ข้อสอบมี 9 หน้า (ไม่รวมใบปะหน้า) แบ่งเป็น 2 ตอน คะแนนรวม 50 คะแนน คิดเป็นคะแนนเก็บ 30 %
- คำตอบทั้งหมดจะต้องเขียนลงในข้อสอบ รวมทั้งเขียนชื่อ รหัสนักศึกษา และ Section ในทุกหน้าของข้อสอบให้ชัดเจน ถ้ากระดาษแผ่นใดไม่มีชื่อ รหัสนักศึกษา และ Section นักศึกษาจะถูกหักคะแนน 0.5 คะแนนต่อแผ่น
- คำตอบส่วนใดอ่านไม่ออก จะถือว่าคำตอบนั้นผิด
- การเขียนโปรแกรม ให้ใช้ภาษา C++
- การเขียนโปรแกรมจะต้องตั้งชื่อตัวแปรให้เหมาะสม และมีคอมเมนต์ในจุดสำคัญต่างๆ โดยให้ทั้งหมดเป็นไปตามหลักการเขียนโปรแกรมที่ดี
- อ่านคำสั่งในแต่ละข้อให้เข้าใจก่อนลงมือทำ
- หากข้อใดเขียนตอบไม่พอ ให้เขียนเพิ่มเติมที่ด้านหลังของหน้านั้นเท่านั้น
- **ทุจริตในการสอบ** โทษขั้นต่ำคือปรับตกในรายวิชานี้และพักการเรียนหนึ่งภาคการศึกษา โทษสูงสุดคือไล่ออก

ตอนที่ 1 Basic C++ (25 คะแนน)

1. โปรแกรมด้านล่างนี้จะแสดงเมนู 1-5 และรอรับค่าจากผู้ใช้ โดยผู้ใช้จะต้องใส่ค่าตัวเลขที่ตรงกับเมนูใดเมนูหนึ่ง ถ้าผู้ใช้ใส่ค่าตัวเลขนอกเหนือไปจากในเมนู โปรแกรมก็จะวนรับค่าไปเรื่อยๆ จนกว่าจะถูกต้อง จงเขียนส่วนของโปรแกรมที่ขาดหายไป (3 คะแนน)

```
#include<iostream.h>

void main() {
    int choice;

    cout << "Mailing list menu: << endl;
    cout << " 1. Enter addresses << endl;
    cout << " 2. Delete addresses << endl;
    cout << " 3. Search the list << endl;
    cout << " 4. Print the list << endl;
    cout << " 5. Quit << endl;

    do {
        (1) _____
        (2) _____
    } while ((3) _____);
}
```

ผลการรันโปรแกรม

Mailing list menu:

1. Enter addresses
2. Delete addresses
3. Search the list
4. Print the list
5. Quit

Enter the number of the choice (1-5) : 0

Enter the number of the choice (1-5) : 5

2. จงเขียน Recursive ฟังก์ชัน (ฟังก์ชันที่มีการเรียกตัวเอง) สำหรับหาค่า $F(x)$ ดังนี้

$$F(x) = 0 \quad \text{ถ้า } x \text{ เท่ากับ } 0$$

$$= 1 \quad \text{ถ้า } x \text{ เท่ากับ } 1$$

$$= F(x-1) + F(x-2) \quad \text{ถ้า } x \text{ มากกว่า } 1$$

กำหนดให้ x และ $F(x)$ เป็นจำนวนเต็ม และให้ตั้งชื่อฟังก์ชันว่า fib (3 คะแนน)

- 2.1 จงบอกข้อเสียของการเขียนฟังก์ชันแบบ Recursive เมื่อเทียบกับการเขียนแบบ Iteration (for,while,do-while) (1 คะแนน)

ชื่อ _____ รหัสนักศึกษา _____ Section _____

3. จงเขียนรายละเอียดของฟังก์ชัน copy1 และ copy2 โดยฟังก์ชัน copy1 ก็อปปีค่า s2 ไปให้ s1 โดยใช้การอ้างถึงข้อมูลแบบอาร์เรย์ ([]) ฟังก์ชัน copy2 ก็อปปีค่า s2 ไปให้ s1 โดยใช้การอ้างถึงข้อมูลแบบพอยเตอร์ (*)

```
#include<iostream.h>
void copy1( char *, const char * );
void copy2( char *, const char * );
int main() {
    char string1[10], *string2 = "Hello";
    char string3[3], string4[ ] = "Good Bye";
    copy1( string1, string2);
    cout << "string1 = " << string1 << endl;
    copy2( string3, string4);
    cout << "string3 = " << string3 << endl;
    return 0;
}
void copy1( char *s1, const char *s2)                ( 4 คะแนน)
{
    _____
    _____
    _____
    _____
    _____
    _____
}
void copy2( char *s1, const char *s2)                ( 4 คะแนน)
{
    _____
    _____
    _____
    _____
    _____
    _____
}
}
```

ผลการรันโปรแกรม
string1 = Hello
string3 = Good Bye

4. หลักการทำงานของ Selection Sort สำหรับอาร์เรย์ใดๆเริ่มจากการหาค่าที่มากที่สุดในอาร์เรย์นั้นๆแล้วสลับค่าที่มากที่สุดไปไว้ในอีลีเมนต์สุดท้าย แล้วนำค่าของอีลีเมนต์สุดท้ายมาไว้ ณ ตำแหน่งที่มีค่าสูงสุด จากนั้นทำซ้ำการทำงานนี้ โดยในแต่ละรอบให้ตรวจเฉพาะอีลีเมนต์ที่ยังไม่ได้ทำการสลับไปไว้ตำแหน่งสุดท้ายของรอบนั้นๆ

จากหลักการทำงานของ Selection Sort จงเขียนอัลกอริทึมหรือ flowchart การทำงาน (อย่างไรก็ตามอย่างหนึ่ง) และได้ของฟังก์ชัน void selection_sort (int a[], int n) ที่ทำการเรียงลำดับอาร์เรย์ a จำนวน n อีลีเมนต์ตามหลักการของ Selection Sort

4.1 อัลกอริทึมหรือ Flowchart (อย่างไรก็ตามอย่างหนึ่ง) ของฟังก์ชัน

```
void selection_sort (int a[], int n)
```

(5 คะแนน)

ตอนที่ 2 OOP (25 คะแนน)

ใช้นิยามของคลาสต่อไปนี้เพื่อตอบคำถามข้อ 5.1-5.5

```
class Rational {
public:
    Rational(); // ให้ค่าเริ่มต้นเป็นศูนย์
    Rational( int numerator, int denominator );
    Rational( int numerator ); // ให้ค่า denominator เป็นศูนย์
    Rational( Rational &);
    Rational operator+ (Rational &r);
    Rational operator* (Rational &r);

private:
    // ฟังก์ชันที่ใช้ในการคำนวณลดรูป numerator และ denominator ให้เป็นเศษส่วนอย่างต่ำ
    void reduce();
    int numerator; // เศษ
    int denominator; // ส่วน
};
```

5.1 คลาสด้านบนมี constructor ทั้งหมดกี่ฟังก์ชัน (1 คะแนน)

5.2 ถ้าต้องการลดจำนวน constructor ของคลาส Rational ให้เหลือจำนวนน้อยที่สุดที่ยังครอบคลุมการทำงานเหมือนที่นิยามไว้ในคลาสข้างต้น ต้องประกาศ constructor อย่างไร (2 คะแนน)

5.3 เราจะประกาศฟังก์ชัน reduce เป็นแบบ Constant ได้หรือไม่ เพราะเหตุใด (2 คะแนน)

```
void reduce( ) const;
```

5.4 เขียนรายละเอียดของโอเปอเรเตอร์ + ที่ใช้ในการบวกค่าของ r เข้ากับตัวมันเอง (4 คะแนน)

```
Rational operator+ (Rational &r)
```

```
{ _____  
_____  
_____  
_____  
_____  
_____  
_____  
_____  
_____  
_____  
_____  
_____
```

}

5.5 เขียนรายละเอียดของโอเปอเรเตอร์ * ที่ใช้ในการคูณค่าของ r เข้ากับตัวมันเอง (4 คะแนน)

```
Rational operator* (Rational &r) {
```

```
{ _____  
_____  
_____  
_____  
_____  
_____  
_____  
_____  
_____  
_____  
_____
```

}

5.6 พิจารณาสวนหนึ่งของคลาสด้านล่าง ซึ่งมีจุดผิดพลาดอยู่ 3 จุด จงบอกว่าเป็นจุดใดบ้าง และ
ผิดพลาดเพราะเหตุใด (3 คะแนน)

```
class Foo {  
    private:  
        int delete;  
        double _d = 0.0 ; ;  
    // ...  
    public:  
        void method1() { }  
}
```

5.7 จากส่วนของโปรแกรมด้านล่าง จงเขียนผลลัพธ์ของโปรแกรม พร้อมทั้งบอกว่าทำไมผลลัพธ์
บรรทัดสุดท้ายจึงเป็นเช่นนั้น (3 คะแนน)

```
#include<iostream.h>  
  
class A {  
    public:  
        void print() { cout << "A" << endl; }  
};  
  
class B : public A {  
    public:  
        void print() { cout << "B" << endl; }  
};  
  
A* a1 = new A;  
B* b1 = new B;  
a1->print();  
b1->print();  
a1 = b1;  
a1->print();
```

5.8 จงบอกว่าจะอะไรคือปัญหาของการประกาศค่าตัวแปรด้านล่างและสามารถแก้ไขได้อย่างไร
(3 คะแนน)

```
const float PI = 3.14;
```

```
float *pi = &PI ;
```

5.9 ฟังก์ชันและข้อมูลแบบ private และ protected ของคลาสแม่ (superclass) สามารถสืบทอด
(inherited) ให้กับคลาสลูก (subclass) ได้ ข้อความนี้ถูกต้องหรือไม่ อย่างไร (2 คะแนน)

5.10 ในภาษา C++ ฟังก์ชัน destructor ของคลาสใดๆ สามารถทำการ overloaded ได้ ข้อความ
นี้ถูกต้องหรือไม่ (1 คะแนน)