PRINCE OF SONGKLA UNIVERSITY

FACULTY OF ENGINEERING

Final Examination: ภาคการศึกษาที่ 1 **Academic Year:** 2547

Date: 2 ตุลาคม พ.ศ. 2547 **Time**: 13.30 – 16.30 น.

Subject Number: 240-204 **Room:** R300

Subject Title: Data Structures and Computer Programming Techniques

---

ทุจริตในการสอบ มีโทษขั้นต่ำ คือ ปรับตกในรายวิชาที่ทุจริต และพักการเรียน 1 ภาคการศึกษา

อ่านรายละเอียดของข้อสอบ และคำแนะนำให้เข้าใจก่อนเริ่มทำข้อสอบ

รายละเอียดของข้อสอบ:

เวลา 3 ชั่วโมง (180 คะแนน: 180 นาที)

ข้อสอบมีทั้งหมด 8 หน้า (ไม่รวมหน้านี้) ประกอบด้วยคำถามจำนวน 4 ข้อ

สิ่งที่สามารถนำเข้าห้องสอบได้:

อนุญาต: เอกสาร หนังสือ และเครื่องเขียนต่าง ๆ

(ให้นักศึกษานำข้อสอบออกไปจากห้องสอบได้)

คำแนะนำ:

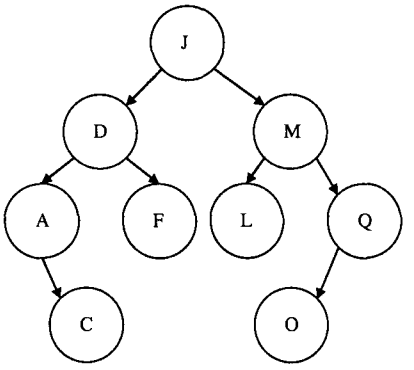
- พยายามทำทุกข้อ
- คำตอบทั้งหมดจะต้องเขียนลงในช่องว่างที่เว้นไว้ในข้อสอบ
- คำตอบส่วนใดอ่านไม่ออก จะถือว่าคำตอบนั้นผิด
- อ่านคำสั่งในแต่ละข้อให้ชัดเจนว่า เขียนโปรแกรมบางส่วน เขียนฟังก์ชัน หรือเขียนทั้งโปรแกรม รวมไปถึงข้อกำหนดเพิ่มเติม และหมายเหตุในข้อนั้น ๆ
- การเขียนโปรแกรมในแต่ละข้อ อาจจะไม่ต้องเขียนตามคำสั่งย่อยทั้งหมด แต่คะแนนจะลดลงตามส่วน และหากในข้อใหญ่หนึ่งข้อ นักศึกษาไม่สามารถทำข้อย่อยข้อแรก ๆ ได้ นักศึกษาสามารถทำข้อย่อยหลัง ๆ โดยให้อ้างอิงเหมือนนักศึกษาทำข้อย่อยข้อแรก ๆ ได้
- การเขียน code จะต้องตั้งชื่อตัวแปรให้เหมาะสม และมี comment ในจุดสำคัญต่าง ๆ โดยให้ทั้งหมดเป็นไปตามหลักการเขียนโปรแกรมที่ดี

ข้อที่ 1 TREE *(25 คะแนน: 25 นาที)*

1.1 จากลำดับข้อมูลดังต่อไปนี้ จงวาดต้นไม้แบบ Binary Search Tree

41 56 78 32 12 20 68 9

1.2 จากต้นไม้แบบ Binary Search Tree ข้างล่างนี้ จง เขียนผลการเยี่ยม node ต่างๆ แบบ Preorder, Postorder และ Inorder



ชื่อ-สกุล _____ รหัสนักศึกษา _____

ข้อที่ 2 EXCEPTION *(35 คะแนน: 35 นาที)*

จากฟังก์ชัน Factorial ที่ให้มาข้างล่างนี้

```
int Factorial( int number )
{
      // return result of factorial n
      if (number == 0)
          return 1;
      else
       return n * Factorial n;
}  // end function Factorial
```

2.1 จงเขียนคลาส `LessThanZero` ที่สืบทอดมาจาก คลาส exception เพื่อเตรียมข้อความแสดงสาเหตุของ exception

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

2.2 จงแสดงฟังก์ชัน `Factorial` ที่ปรับปรุงเพิ่มส่วนของการตรวจสอบพารามิเตอร์ `n` ของฟังก์ชัน `Factorial` ว่ามีค่าน้อยกว่า 0 หรือไม่ และใช้คำสั่ง `throw` เมื่อพบ exception ดังกล่าว

.....

.....

2.3 จง แสดงโปรแกรมทดสอบ ที่ใช้คำสั่ง `try` และ `catch`

ข้อที่ 3 FILE & STRING & STACK ***(40 คะแนน: 40 นาที)***

จงเขียนโปรแกรมอย่างสมบูรณ์เพื่ออ่านไฟล์ (Text-File) ชื่อ palin.txt เพื่อที่จะตรวจนับว่าในไฟล์ดังกล่าวมีคำที่เป็น Palindrome กี่คำ โดยให้รายงานผลของจำนวนคำ และคำที่ปรากฏในตอนท้ายของโปรแกรม

นักศึกษาจะต้องมีการใช้งาน Stack เพื่อตรวจจับ Palindrome

ข้อที่ 4 *(80 คะแนน: 80 นาที)*

จากส่วนของโปรแกรมที่กำหนดให้ใน ฟังก์ชัน `main()` ที่กำหนดให้ข้างล่าง ซึ่งมีการสร้างอาเรย์ของพอยต์เตอร์ Shape ที่ภายในมีการบรรจุพอยต์เตอร์ชนิด `Circle` และ `Rectangle` (class `Circle` และ `Rectangle` สืบทอดมาจาก class `Shape` โดยให้อ่านรายละเอียดได้จากข้อ 4.4)

```
#include <iostream>
#include <list>
#include <map>

#include "shape.h"

int main()
{
  Shape * s_ptrs[3];
  s_ ptrs[0] = new Circle( 10 );
  s_ ptrs[1] = new Rectangle( 4, 5 );
  s_ ptrs[2] = new Rectangle( 3, 2 );

  /*Q4.1*/

  /*Q4.3*/

  for(int i = 0; i < 3; i++)
      delete s_ ptrs[i];

  return 0;
}
```

4.1 นักศึกษาเขียนส่วนของโปรแกรมที่ขาดหายไปในฟังก์ชัน `main()` เพื่อนำสมาชิกทั้งหมดใน `s_ptrs` ไปใส่ใน vector แล้วทำการหาค่าผลรวมพื้นที่ของทุก Object ภายใน vector

ทั้งนี้ในการหาค่าผลรวมของพื้นที่จะต้องมีการเรียกใช้ฟังก์ชัน `getTotalArea` อย่างเหมาะสม (การนิยามฟังก์ชัน `getTotalArea` ให้ทำการนิยามในข้อที่ 4.2) (20 คะแนน)

4.2 จงเขียนส่วนการนิยามฟังก์ชัน `getTotalArea` (20 คะแนน)

4.3 จงใช้งาน map อย่างเหมาะสม เพื่อพิพม์รายละเอียดของ Object เรียงลำดับตามพื้นที่ของ Object นั้น ๆ ทางหน้าจอ (25 คะแนน)

4.4 จงปรับปรุงส่วนของ Code สำหรับ class Shape, Circle และ Rectangle อย่างเหมาะสม เพื่อให้สามารถทำงานได้ในข้อ 4.1 และ 4.2 อย่างเหมาะสม โดยให้นักศึกษาแทรก แก้ไข หรือลบได้บนโค้ดที่กำหนดให้ (15 คะแนน)

หมายเหตุ ในข้อนี้ อนุญาตให้นักศึกษาใช้ปากกาเท่านั้น เพื่อความสะดวกในการตรวจ

```
#include <iostream>

class Shape
{
public:
  float getArea()
  {
    return 0;
  };

  void print();
};
```

```
class Rectangle : public Shape
{
private:
  int w, h;

public:
  Rectangle( int w, int h ) : w( w ), h( h )
  {
  };

  float getArea()
  {
    return w * h;
  };

  void print()
  {
    std::cout << "Rectangle [" << w << ":" << h << "]" <<
std::endl;
  }
};


class Circle : public Shape
{
private:
  int r;

public:
  Circle( int r ) : r( r )
  {
  };

  float getArea()
  {
    return 3.17 * r * r;
  };

  void print()
  {
    std::cout << "Circle [" << r << "]" << std::endl;
  }
};
```