

PRINCE OF SONGKLA UNIVERSITY
FACULTY OF ENGINEERING



Final Examination: ภาคการศึกษาที่ 2

Academic Year: 2547

Date: 28 กุมภาพันธ์ พ.ศ. 2547

Time: 09:00 -12:00

Subject Number: 240-207

Subject Title: Intro to Software Eng and Database Sys

ทฤษฎีในการสอบ มีโทษขั้นต่ำ คือ ปรับตกในรายวิชาที่ทฤษฎี และพักการเรียน 1 ภาคการศึกษา

อ่านรายละเอียดของข้อสอบ และคำแนะนำให้เข้าใจก่อนเริ่มทำข้อสอบ

รายละเอียดของข้อสอบ:

เวลา 3 ชั่วโมง (180 คะแนน: 180 นาที)

ข้อสอบมีทั้งหมด 3 หน้า (ไม่รวมหน้านี้) ประกอบด้วยคำถามจำนวน 6 ข้อ

สิ่งที่สามารถนำเข้าห้องสอบได้:

อนุญาต: เครื่องเขียนต่าง ๆ เช่น ปากกา หรือดินสอ

ไม่อนุญาต: หนังสือ และเครื่องคิดเลข

คำแนะนำ:

- พยายามทำทุกข้อ
- คำตอบทั้งหมดจะต้องเขียนในสมุดคำตอบที่แจกมาพร้อมกับข้อสอบ
- คำตอบอาจไม่เรียงตามลำดับข้อ แต่จะต้องเขียนข้อกำกับให้ชัดเจน แต่ข้อย่อยที่อยู่ภายในข้อใหญ่เดียวกันจะต้องอยู่ด้วยกัน และทุกครั้งที่ขึ้นข้อใหม่ จะต้องขึ้นหน้าใหม่ทุกครั้ง
- คำตอบส่วนใดอ่านไม่ออก จะถือว่าคำตอบนั้นผิด
- อ่านคำสั่งในแต่ละข้อให้ชัดเจนว่า ให้วาดรูปโดยสมบูรณ์หรือวาดโดยไม่ต้องแสดงรายละเอียด
- การทฤษฎีในห้องสอบ อาจมีโทษถึง ไล่ออก

ข้อที่ 1 SOFTWARE ENG. & DATABASE**(20 คะแนน: 20 นาที)**

1.1 จงใช้ข้อความเพียงหนึ่งบรรทัด เพื่ออธิบายว่าเหตุใดเราจึงจำเป็นต้องเรียนรู้กระบวนการทาง Software Engineer (5)

1.2 จงใช้ Diagram 1 อัน พร้อมข้อความสั้น ๆ เพื่ออธิบายว่ากระบวนการทาง Software Engineer สัมพันธ์กับกระบวนการทาง Database Design อย่างไร (15)

ข้อที่ 2 SOFTWARE LIFE CYCLE**(30 คะแนน: 30 นาที)**

2.1 จงเติมตารางข้างล่างให้สมบูรณ์ โดยให้ใส่เครื่องหมาย / ในช่องของ SDLC ที่มีคุณสมบัติตรงกับที่ระบุไว้ในด้านซ้ายมากที่สุด และให้ใส่เครื่องหมาย X ในช่องที่มีคุณสมบัตินั้นน้อยที่สุด

	Waterfall	Prototyping	Evolutionary Models	XP Programming
Coping with Change				
Involving the user				
Good Documentation				
Overcoming technical difficulties along the way				
Structured Code				

2.2 หากต้องการพัฒนาซอฟต์แวร์ขนาดกลาง และมีข้อจำกัดอยู่ว่าจะต้องมีความผิดพลาดของซอฟต์แวร์น้อยที่สุดเมื่อมีการนำไปใช้งาน ควรจะใช้ SDLC แบบใด ให้เหตุผลประกอบ

2.3 จงอธิบายว่า ในสถานการณ์ใดที่เราจำเป็นต้องใช้เทคนิคของ Reverse Engineering พร้อมทั้งอธิบายโดยสังเขปถึงกระบวนการในการทำ Reverse Engineering (อาจใช้ตัวอย่างประกอบ)

ข้อที่ 3 REQUIREMENTS**(40 คะแนน: 40 นาที)**

3.1 จงยกตัวอย่าง Non-functional requirement มา 1 ชนิด พร้อมทั้งบอกว่า จะเขียน requirement นั้นอย่างไรเพื่อให้ requirement มีคุณสมบัติเป็น requirements verifiability (10)

3.2 จากประโยค

“การศึกษา Requirement อย่างถี่ถ้วนถือเป็นการลดความเสี่ยง (Risk) ได้อย่างมาก”

จงแสดงความคิดเห็นต่อประโยคข้างต้นว่าเห็นด้วยหรือไม่ อย่างไร (10)

3.3 บริษัทแห่งหนึ่งต้องการพัฒนาระบบเก็บเอกสาร โดยมี Requirements ดังต่อไปนี้

ผู้ใช้สามารถที่จะสร้างเอกสารของตนไว้ในเซิร์ฟเวอร์ได้ 2 วิธีด้วยกัน

วิธีที่ 1 สร้างเอกสารที่มีเฉพาะข้อความโดยใช้ *text editor* ผ่านระบบ

วิธีที่ 2 สร้างโดยการ *upload* ไฟล์ขึ้นสู่ระบบ โดยไฟล์จะต้องอยู่ในรูปแบบของ *PDF*

เจ้าของเอกสารสามารถที่จะเลือกเอกสารจากรายการเอกสาร เพื่อที่จะทำการอ่านเอกสาร หรือ ส่งเอกสารให้กับผู้ใช้คนอื่น เพื่อที่จะให้ผู้ใช้ที่ได้รับเอกสารสามารถเปิดอ่านเอกสารนั้นได้ด้วยเช่นกัน อย่างไรก็ตามระบบจะไม่อนุญาตให้ผู้ใช้รับเอกสารส่งเอกสารต่อไปยังผู้อื่น โดยในการส่งนั้นหากเอกสารเป็นเอกสารประเภท *PDF* เจ้าของสามารถระบุหมายเลขหน้าที่ต้องการส่งได้ อย่างไรก็ตามเพื่อเป็นการประหยัดเนื้อที่ในการจัดเก็บ ระบบจะบันทึกเพียงหมายเลขหน้า โดยจะทำการตัดต่อเอกสารเป็นครั้งคราวเมื่อผู้ใช้เป็นผู้รับเอกสารเข้ามาอ่านเอกสาร

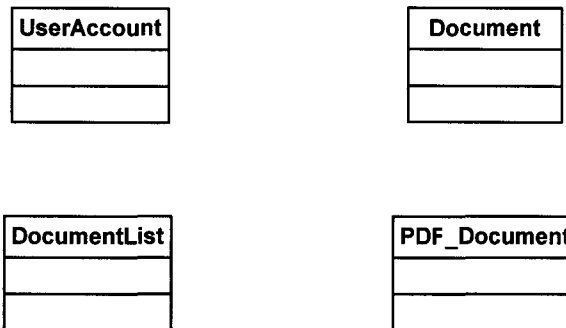
จงวาด Use-case diagram ที่สัมพันธ์กับ Requirements ข้างต้น (20)

ข้อที่ 4 CLASS & SEQUENCE DIAGRAMS

(40 คะแนน: 40 นาที)

จาก Requirements ในข้อ 3.2

4.1 จากโครงของ Class Diagram ที่กำหนดให้ จงเติม Diagram ให้สมบูรณ์



4.2 จงวาด Sequence Diagram ในช่วงของการสร้างและส่งเอกสารประเภท PDF

ข้อที่ 5 DESIGN QUALITY**(20 คะแนน: 20 นาที)**

“ในการพัฒนาซอฟต์แวร์ที่ดี ควรจะถูกแบ่งออกเป็น Modules และให้ module เหล่านั้นทำงานประสานกัน”

5.1 จงบอกว่า คุณสมบัติภายใน Module ที่ดีควรมีอยู่ในระดับสูง คืออะไร

5.2 จงบอกว่า คุณสมบัติระหว่าง Module ที่ดีควรมีอยู่ในระดับต่ำ คืออะไร

ข้อที่ 6 TESTING**(30 คะแนน: 30 นาที)**

6.1 จาก Code ดังกล่าว จงวาด flow graph พร้อมทั้ง หาค่าของ Cyclomatic Complexity (CC)
(20)

```
public boolean find(int key) { //Binary Search
    int bottom = 0;
    int top = _elements.length-1;
    int lastIndex = (bottom+top)/2;
    int mid;
    boolean found = key == _elements[lastIndex];

    while ((bottom <= top) && !found) {
        mid = (bottom + top) / 2;
        found = key == _elements[mid];
        if (found) {
            lastIndex = mid;
        } else {
            if (_elements[mid] < key) {
                bottom = mid + 1;
            } else { top = mid - 1; }
        }
    }
    return found;
}
```

6.2 จากค่า CC ในข้อที่ 6.1 จงวิเคราะห์ว่า module ข้างต้นเป็น module ที่ดีหรือไม่ ให้เหตุผล (10)