



มหาวิทยาลัยวงจกรนครินทร์  
คณะวิศวกรรมศาสตร์

สอบกลางภาค ภาคการศึกษาที่ 2

ปีการศึกษา 2550

วันที่สอบ 27 ธันวาคม 2550

เวลาสอบ 13.30 – 15.30 น.

ห้องสอบ A 201, A205

ผู้สอน ดร.อารีย์ ธีรภาพเสรี

รหัสและชื่อวิชา: 240-207 Introduction to Software Engineering and Database system

ทุจริตในการสอบ มีโทษขั้นต่ำ คือ ปรับตกในรายวิชาที่ทุจริต และพักการเรียน 1 ภาคการศึกษา

คำสั่ง: อ่านรายละเอียดของข้อสอบ และคำแนะนำให้เข้าใจก่อนเริ่มทำข้อสอบ

อนุญาต: เครื่องเขียนต่างๆ เช่น ปากกา หรือดินสอ เข้าห้องสอบ

ไม่อนุญาต: หนังสือ และเอกสารใดๆ เข้าและออกห้องสอบ

ข้อสอบมี 13 หน้า (รวมใบปะหน้า) แบ่งเป็น 2 ตอน ตอนที่ 1 มี 40 ข้อ ตอนที่ 2 มี 6 ข้อ

คิดเป็นคะแนนเก็บ 35 %

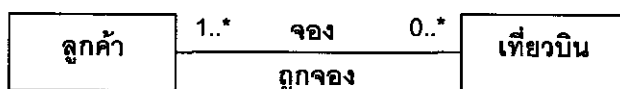
เวลา: 2 ชั่วโมง (120 นาที)

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

ตอนที่ 1 ตอบคำถามต่อไปนี้ โดยเขียนตัวเลือกในตารางที่ให้ไว้ในหน้าที่ 11 (ท้ายข้อที่ 40)

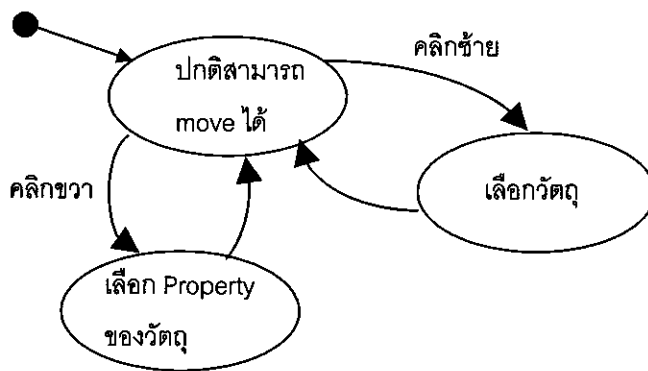
1. ข้อใดต่อไปนี้ไม่ถูกต้องเกี่ยวกับ Software Reengineering
  - ก. เพื่อให้ Software เข้ากับ Hardware ใหม่ที่มีความทันสมัย
  - ข. ค่าใช้จ่ายของการทำ Software Reengineering จะถูกกว่าการทำ Software ใหม่
  - ค. การทำ Software Reengineering ทำให้เราเข้าใจลูกค้ามากขึ้นเนื่องจากต้องถามความต้องการจากลูกค้าใหม่ทั้งหมด
  - ง. Software Reengineering ทำให้ Software เดิมมีประสิทธิภาพขึ้น
  - จ. Software Reengineering มีความเสี่ยงน้อยกว่าการทำ Software ใหม่
2. การพัฒนาซอฟต์แวร์ในแบบ XP และ Reengineering แตกต่างกันอย่างใด
  - ก. XP มีการทดสอบไปพร้อมๆกับการ Coding แต่ Reengineering ไม่ทำ
  - ข. Reengineering มีการใช้งบประมาณและมีความเสี่ยงในการพัฒนามากกว่าแบบ XP
  - ค. Reengineering มีลูกค้าเป็นส่วนหนึ่งของระบบการพัฒนาแต่ XP ไม่มี
  - ง. ขั้นตอนแรกของการพัฒนาซอฟต์แวร์ XP จะทำการของ Requirement แต่ระบบ Reengineering มีการศึกษาโครงสร้างและรายละเอียดงานเดิม
  - จ. ถูกทุกข้อ
3. ข้อใดถูกต้องเกี่ยวกับ Requirement
  - ก. บอกว่าระบบสามารถที่จะทำงานในส่วนใดได้บ้างในขั้นตอนการพัฒนา
  - ข. บอกว่าบริการของระบบจะประกอบไปด้วย Code อะไรบ้าง
  - ค. อธิบายว่าระบบมีบริการใดบ้างจากความต้องการของลูกค้า
  - ง. อธิบายว่าระบบมีความน่าเชื่อถือเพียงใด มี budget เท่าไร ความเร็วเท่าไร
  - จ. อธิบายการทำงานของโปรแกรมทั้งหมด เพื่อให้ลูกค้าใช้เป็นอ้างอิงในการใช้งาน
4. ข้อใดคือหลักในการเขียน Requirements definition
  - ก. เมื่อติดต่อพูดคุยกับลูกค้าและนำความต้องการของลูกค้าที่ได้มาเขียนแล้วนำกลับไปให้ลูกค้าตรวจสอบว่าตรงตามที่ต้องการหรือไม่และนำกลับมาแก้ไข เขียนด้วยภาษาธรรมดาให้ลูกค้าอ่านแล้วเข้าใจ
  - ข. เมื่อติดต่อพูดคุยกับลูกค้าแล้วนำความต้องการของลูกค้ามาเขียนเป็นโครงสร้างของระบบมีส่วนของ interface และรายละเอียดทางเทคนิคต่างๆ
  - ค. เขียนเกี่ยวกับรายละเอียดของ Software ที่จะนำมาใช้ในการพัฒนาระบบ
  - ง. นำความต้องการที่ได้จากลูกค้ามาเขียนแยกเป็นส่วนๆและแต่ละส่วนจะมีฟังก์ชันอะไรบ้างที่ใช้ในการทำงาน
  - จ. เขียน Prototyping ให้ลูกค้าดูคร่าวๆ
5. ข้อใดต่อไปนี้ เป็นความต้องการแบบ non-function requirement
  - ก. ข้อมูลมีการอัปเดต
  - ข. เวลาที่ผู้ใช้สามารถอยู่ในระบบได้คือไม่เกิน 60 นาที
  - ค. ถ้าลูกค้าใส่รหัสผิด 3 ครั้งติดกันเครื่องจะยึดบัตร
  - ง. ลูกค้าสามารถขอรายละเอียดของค่าใช้จ่ายได้
  - จ. ข้อ ข และ ค ถูก

6. บริษัททำซอฟต์แวร์แห่งหนึ่งได้รับการจ้างให้ทำโปรแกรมจัดทำบัญชีของบริษัททางการเงินแห่งหนึ่งซึ่งมีผู้ที่สามารถตัดสินใจเรื่องรูปแบบโปรแกรมหลายคน ปัญหาใดน่าจะเกิดมากที่สุด
- ไม่สามารถทำได้เรื่องปัญหาทางด้านเทคนิค
  - ปัญหาเรื่องภาษา
  - ผู้ใช้หลายคนต้องการ requirement หลายแบบ
  - ลูกค้าหลายคนต้องการ requirement ที่แตกต่างกัน
  - การตัดสินใจของโปรแกรมเมอร์
7. Requirements แบบใดที่เป็น Requirements ที่จัดทำขึ้นสำหรับ Customers
- Requirements definition และ Software specification
  - Requirements specification และ Software specification
  - Software specification และ Requirements document
  - Requirements definition และ Requirements specification
  - Requirements definition และ Requirements Verifiability
8. ข้อใดกล่าวไม่ถูกต้องเกี่ยวกับ Use Case diagram
- Use Case diagram เป็นการแสดงความสัมพันธ์ระหว่าง Actor และ Use Case
  - ใน UML จะใช้วงรีแทน Use Case
  - Actors จะใช้รูปคน ซึ่งจะหมายถึงคนที่ทำงานในระบบ
  - Use Case ถูกจำกัดโดย Users หรือระบบอื่นที่เรียกว่า Actors
  - Use Case diagram เป็นเครื่องมือเริ่มต้นที่ใช้ในการค้นหา Requirement ของระบบ
9. ข้อใดอธิบายรูปแบบความสัมพันธ์ได้ถูกต้อง เกี่ยวกับความสัมพันธ์ระหว่าง Use Case
- ความสัมพันธ์แบบ Include คือความสัมพันธ์ที่ Use Case หนึ่งเรียกใช้อีก Use Case หนึ่งแบบอัตโนมัติ
  - ความสัมพันธ์แบบ Extend คือความสัมพันธ์ที่ Use Case แ่งฟังก์ชันให้มีความพิเศษ
  - ความสัมพันธ์แบบ Generalization คือความสัมพันธ์ที่ Use Case นี้จะทำงานอัตโนมัติ
  - ความสัมพันธ์แบบ Include คือความสัมพันธ์ที่ตายตัว มีรูปแบบที่แก้ไขไม่ได้
  - ความสัมพันธ์แบบ Extend คือความสัมพันธ์ที่อ้างอิงถึงอีกรูปแบบหนึ่งแบบมีเงื่อนไข
10. จากรูปข้อใดอธิบายความสัมพันธ์ได้ถูกต้อง



- ลูกค้าจองเที่ยวบินได้อย่างน้อย 1 เที่ยวบิน
- ลูกค้าจองเที่ยวบินได้ก็เที่ยวบินก็ได้ หรือไม่จองก็ได้
- เที่ยวบินถูกจองโดยลูกค้าได้มากกว่า 0
- เที่ยวบินถูกจองโดยลูกค้าได้อย่างน้อย 1
- ลูกค้า 1 คนจองเที่ยวบินได้ก็เที่ยวบินก็ได้ หรือไม่จองก็ได้ เที่ยวบินต้องถูกจองโดยลูกค้าได้อย่างน้อย 1 เที่ยวบิน

11. ความสัมพันธ์แบบ aggregation และ generalization นำมาอธิบายรูปแบบการเขียน Code แบบ OOP ได้อย่างไร
- แบบ ISA ทั้งคู่
  - แบบ HASA ทั้งคู่
  - แบบ HASA และแบบ ISA ตามลำดับ
  - แบบ ISA และแบบ HASA ตามลำดับ
  - ไม่มีข้อใดถูกต้องตามหลัก OOP
12. จากรูปหากสถานะเดิมอยู่ที่ "ปกติสามารถ move ได้" นั้นสามารถเปลี่ยนเป็นสถานะ "เลือกวัตถุ" ได้เมื่อเกิดเหตุการณ์อะไร จาก state diagram ด้านล่าง

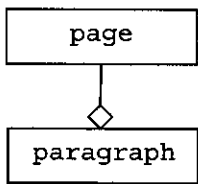


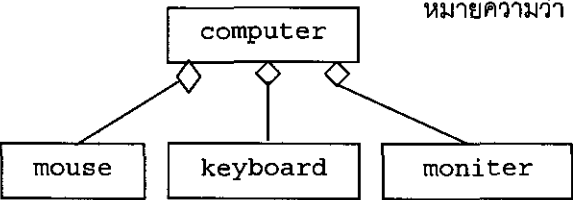
- คลิกขวา
  - คลิกซ้าย
  - ดับเบิ้ลคลิก
  - ลากเมาส์
  - ทริปปเบิลคลิก
13. ข้อใดคือความหมายของ Inheritance
- การสืบทอดคุณสมบัติจาก Parent class มายัง Child class คุณสมบัติพื้นฐานจะเหมือนกัน และมีคุณสมบัติ ของ Child class เพิ่มขึ้น
  - การสืบทอดคุณสมบัติบางส่วนจาก Parent class มายัง Child class
  - การสืบทอดโดยการคัดลอก Parent class มายัง Child class
  - การแยกตัว Child class ออกจาก Parent class
  - ข้อ ข และ ง ถูกต้อง
14. ข้อใดกล่าวถึง state diagram ได้ถูกต้อง
- การอธิบายสถานะการทำงานของ class diagram ทั้งหมด
  - การอธิบายสถานะการทำงานของ class ใด class หนึ่งใน class diagram เท่านั้น
  - การอธิบายการทำงานระหว่าง class สอง class เท่านั้น
  - สถานะการทำงานของ software ที่เราจะสร้าง
  - การแสดงความสัมพันธ์ของ class สอง class

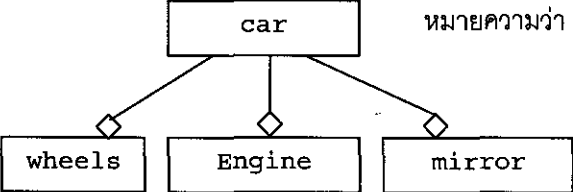
15. ข้อใดผิด

- ก. Use Case diagram เป็น diagram ที่ทำหน้าที่หา requirements
- ข. Use Case ทำหน้าที่อธิบายว่า ระบบทำงานอะไร
- ค. Use Case Diagram เป็นการแสดงความสัมพันธ์ระหว่าง Actor และ Use Case
- ง. Use Case Models อธิบายพฤติกรรมภายในระบบ
- จ. Actor เป็นคนหรือกลไกอื่น ๆ ที่อยู่ภายนอกระบบ แต่มีความสัมพันธ์กับระบบหรือมีอิทธิพลต่อระบบในรูปแบบใดรูปแบบหนึ่ง

16. ข้อใดอธิบายความหมายของ Aggregation Association ได้ถูกต้อง

ก.  หมายความว่าในหน้ากระดาษ 1 หน้ากระดาษ ประกอบไปด้วย paragraph

ข.  หมายความว่า คอมพิวเตอร์ประกอบด้วย เมาส์, คีย์บอร์ดและจอภาพ

ค.  หมายความว่า รถยนต์ประกอบด้วย ล้อรถ, เครื่องยนต์และกระจก

- ง. ไม่มีข้อถูก
- จ. ถูกทุกข้อ

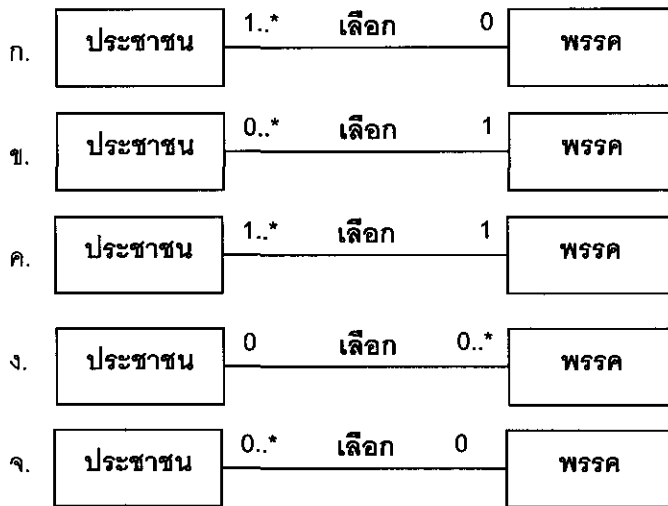
17. Diagram ที่ประกอบด้วยสถานะต่างๆของ Object และเหตุการณ์ต่างๆที่ทำให้สถานะของ Object เปลี่ยน และการกระทำที่เกิดขึ้นเมื่อสถานะของระบบเปลี่ยนไปเรียกว่าอะไร

- ก. Collaboration Diagram
- ข. Use Case Diagram
- ค. State Diagram
- ง. Sequence Diagram
- จ. Class Diagram

18. ข้อใดแสดงให้เห็นถึงข้อมูลที่ทำให้เกิดการเปลี่ยนแปลงใน Software

- ก. Use Case diagram
- ข. Dynamic Modeling
- ค. Class Diagram
- ง. Static behavior
- จ. Activities

19. ในการเลือกตั้งประชาชน 1 คน จะต้องเลือกพรรคการเมือง 1 พรรค แต่ละพรรคการเมืองสามารถมีคนมาเลือกได้ไม่จำกัด จากที่กล่าวข้างต้น จะเขียนความสัมพันธ์ได้ดังข้อใด



20. Scenario คืออะไร

- ก. ความสัมพันธ์ใน Use Case Diagram
- ข. กิจกรรมที่ดำเนินตามปกติ
- ค. ลำดับกิจกรรมที่เกิดขึ้น
- ง. กิจกรรมใน Use Case ที่ผิดปกติ
- จ. Flow of events ที่เป็นไปได้ทั้งหมด

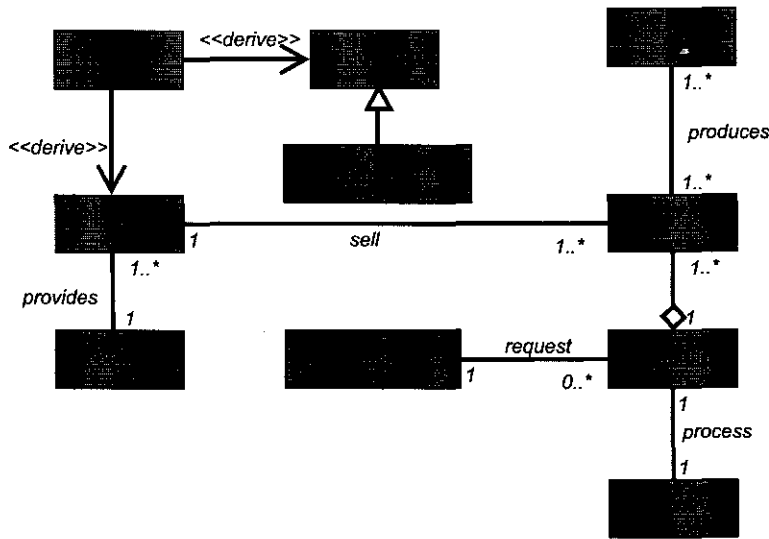
21. ข้อใดต่อไปนี้กล่าวไม่ถูกต้องเกี่ยวกับ Software Design Quality

- ก. การออกแบบระบบที่ดีควรออกแบบให้ระบบสามารถรองรับเทคโนโลยีใหม่ๆได้
- ข. หากระบบมีการติดต่อสื่อสารกันมากควรออกแบบแบบ functional design
- ค. การออกแบบระบบแบ่งเป็น 2 ส่วนคือ Interface กับ Implement
- ง. การแยกซอฟต์แวร์เป็นส่วนๆ ทำให้ปัญหาจัดการได้ง่ายขึ้น
- จ. Object-oriented design จะเน้นว่ามี Object ใดอยู่ในระบบบ้าง

22. ข้อใดไม่ใช่หลักการออกแบบที่ดีตามหลักของ Design Quality

- ก. ออกแบบให้ซอฟต์แวร์มีลักษณะเป็นหน่วยย่อยๆ
- ข. ออกแบบให้แต่ละหน่วยย่อยเป็นอิสระต่อกัน
- ค. ทำซ้ำตามขบวนการเป็นขั้นตอนไปเรื่อยๆ โดยพิจารณาตาม Requirement ที่ต้องการ
- ง. แยกส่วนของข้อมูลและส่วนของฟังก์ชันออกจากกัน
- จ. ไม่จำเป็นต้องมีตัวแปรที่แสดงถึงค่าของข้อมูลที่แยกออกจากกันอย่างชัดเจน

23. จาก Class Diagrams ที่กำหนด ข้อใดสื่อความหมายของ Class Diagrams ได้ถูกต้องที่สุด



- ก. ลูกค้า (Retail Customer) สามารถซื้อสินค้า หรือสินค้าถูกขายให้ลูกค้า (Selling) ได้มากกว่าหนึ่งครั้ง ดังนั้น Class ทั้งสองจึงมีความสัมพันธ์กันแบบ 1-M Association
- ข. การขายสินค้า (Selling) แต่ละครั้งจะถูกดำเนินการโดยผู้ขายสินค้า (Seller) โดย Seller หนึ่งคนสามารถขายสินค้าได้เพียงหนึ่งรายการ จึงกำหนดให้ Class ทั้งสองสัมพันธ์กันแบบ 1-1 Association
- ค. ผู้ผลิตสินค้า (Producer) หนึ่งบริษัท ผลิตสินค้า (Goods) ได้เพียงประเภทเดียว ในขณะที่เดียวกันสินค้าหนึ่งประเภทสามารถผลิตได้โดยหลายบริษัท ดังนั้นทั้งสอง Class มีความสัมพันธ์แบบ M-1 Association
- ง. การสั่งสินค้าโดยลูกค้า (Ordering) แต่ละครั้ง จะมีสินค้า (Goods) ที่ประกอบอยู่ในรายการจำนวนหนึ่งอย่างย่อย ศูนย์รายการ ดังนั้น Class ทั้งสองมีความสัมพันธ์แบบ M-M aggregation
- จ. ลูกค้า (Retail Customer) สามารถสั่งสินค้า (Ordering) ไม่สั่งสินค้า หรือสั่งได้มากกว่าหนึ่งครั้ง ดังนั้น Class ทั้งสองจึงมีความสัมพันธ์แบบ 1-1 Association

24. ข้อใดเป็นการ test ในระดับ function

- ก. Unit testing
- ข. Sub-system testing
- ค. System testing
- ง. Acceptance testing
- จ. Data testing

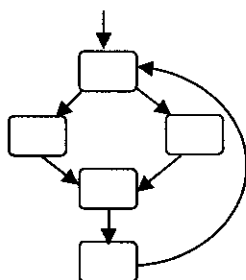
25. ข้อใดเป็นเทคนิคในการ test ระดับ System testing

- ก. Black box testing
- ข. Glass box testing
- ค. White box testing
- ง. Gray box testing
- จ. Defect testing

26. ข้อใดอธิบายเกี่ยวกับ Testing process ได้ถูกต้อง

- ก. Unit testing คือการ test ในระดับฟังก์ชัน
- ข. Sub-system testing คือการรวมกลุ่ม class และ function ที่ทำงานด้วยกันมา test
- ค. System testing คือ ดูว่าระบบสามารถทำงานตรงตามที่เรากำหนดไว้หรือไม่
- ง. Acceptance testing คือการนำระบบไปให้ลูกค้าทดสอบ
- จ. ถูกทุกข้อ

27. จาก Flow graph ต่อไปนี้จงหาค่า Complexity



- ก. 2
- ข. 3
- ค. 4
- ง. 5
- จ. 6

28. จงเลือกข้อที่ถูกที่สุดที่เป็นการทดสอบแบบ Black Box

1. การทดสอบโดยไม่คำนึงถึงคำสั่งภายในโปรแกรม
2. การทดสอบโดยสนใจเฉพาะภายในโปรแกรม
3. การทดสอบฟังก์ชันต่างๆ ของโปรแกรมตาม Requirements ที่มี
4. การทดสอบโดยดูค่า Output จาก Input ที่มีความสัมพันธ์กัน
5. การทดสอบโดยดูค่า Input จาก Output ที่มีความแตกต่างกัน

- ก. ข้อ 2, 3, 5 ถูกต้อง
- ข. ข้อ 2, 3, 4 ถูกต้อง
- ค. ข้อ 1, 3, 4, 5 ถูกต้อง
- ง. ข้อ 1, 3, 4 ถูกต้อง
- จ. ข้อ 1, 4, 5 ถูกต้อง

29. ข้อใดเรียงตามลำดับของ testing process

- ก. Unit testing → Sub-system testing → System testing → Acceptance testing
- ข. Unit testing → System testing → Sub-system testing → Acceptance testing
- ค. Sub-system testing → System testing → Acceptance testing → Unit testing
- ง. System testing → Acceptance testing → Unit testing → Sub-system testing
- จ. Acceptance testing → Unit testing → Sub-system testing → System testing



30. ข้อใดไม่ใช่เทคนิคการประเมินราคาของซอฟต์แวร์ที่ควรทำ
- ก. ประเมินราคาจากการดูจากโปรเจ็คเก่าๆที่เคยทำมา
  - ข. ประเมินจากสภาพความคล่องตัวของเศรษฐกิจ
  - ค. ใช้ผู้ชำนาญการมาประเมินราคา
  - ง. ใช้สูตรทางคณิตศาสตร์ในการประเมินราคา
  - จ. อ้างอิงจากงบประมาณของลูกค้า
31. ข้อใดคือความหมายของ Pricing to win
- ก. ใช้เงินอย่างชาญฉลาด
  - ข. ให้เวลาอย่างคุ้มค่า
  - ค. ให้แรงงานอย่างเหมาะสม
  - ง. ประเมินราคาตามงบประมาณของลูกค้า
  - จ. ได้กำไรจากลูกค้าอย่างมาก
32. ข้อใดไม่ได้เป็นปัจจัยของการตั้งราคาสินค้า
- ก. ปัจจัยการตลาด
  - ข. การเป็นผู้ถือลิขสิทธิ์ของซอฟต์แวร์
  - ค. การแข่งขันระหว่างบริษัท
  - ง. การเงินของบริษัท
  - จ. จำนวนพนักงานในบริษัท
33. ข้อใดเป็นลักษณะของ Organic Project
- ก. โปรเจ็คขนาดใหญ่ ใช้ทีมงานเยอะ
  - ข. โปรเจ็คขนาดเล็ก ความต้องการของระบบยืดหยุ่น ใช้ทีมงานไม่มาก
  - ค. โปรเจ็คขนาดเล็ก ความต้องการของระบบไม่ยืดหยุ่น
  - ง. โปรเจ็คขนาดใหญ่ ความต้องการของระบบยืดหยุ่น
  - จ. ข้อ ข และ ค ถูกต้อง
34. ข้อใดไม่ใช่ปัญหาของการจัด Schedule ของโปรเจ็ค
- ก. ความยากของโปรเจ็ค
  - ข. ทีมพัฒนามีความสามารถไม่เพียงพอ
  - ค. การเพิ่มโปรแกรมเมอร์ในทีมพัฒนา
  - ง. การลดจำนวนคนที่มีประสิทธิภาพสูงลง
  - จ. การจัดส่งล่าช้าของตัวอุปกรณ์ที่สั่งซื้อมาใช้งานในโปรเจ็ค
35. คำกล่าวในข้อใดถูกต้อง
- ก. COCOMO Model เป็นโมเดลที่ใช้ประเมินราคา Software ที่ใช้งานบ่อยที่สุด
  - ข. Expert Judgment เป็นวิธีประเมินราคาซอฟต์แวร์ที่ดีที่สุด
  - ค. จำนวนคนที่เพิ่มขึ้น ทำให้งานเสร็จเร็วขึ้น
  - ง. Top down เริ่มจากการหาว่าระบบมี Module ย่อยๆ อะไรบ้าง

- จ. ภาษา Assembly เป็นภาษาที่ทำให้งานเสร็จเร็วกว่าภาษาอื่น ๆ
36. การคำนวณขนาดของซอฟต์แวร์แบบ Basic COCOMO โปรแกรมใดจัดว่ามีข้อบกพร่องมากที่สุด
- ก. Organic Project
  - ข. Dynamic Project
  - ค. Semi-detached Project
  - ง. Embedded Project
  - จ. Static Project
37. ข้อใดกล่าวถูกต้องในแง่การพัฒนาซอฟต์แวร์
- 1. สิ่งแวดล้อมที่ตี Software ที่ได้จะดีด้วย
  - 2. ความตั้งใจของพนักงาน ทำให้ได้ซอฟต์แวร์ที่มีคุณภาพ
  - 3. ซอฟต์แวร์ที่ไม่ขึ้นกับจำนวนคนในการพัฒนา
  - 4. ซอฟต์แวร์ที่ได้ต้องตรงตาม requirement ที่ได้กำหนดไว้
- ก. ข้อ 1 และ 2 ถูกต้อง
  - ข. ข้อ 3 และ 4 ถูกต้อง
  - ค. ข้อ 1, 2 และ 3 ถูกต้อง
  - ง. ข้อ 1, 3 และ 4 ถูกต้อง
  - จ. ข้อ 1 ถึง 4 ถูกต้อง
38. งานหลักของ Software Project Manager คือข้อใด
- ก. คัดเลือกทีมพัฒนาซอฟต์แวร์
  - ข. การวางแผนการพัฒนาซอฟต์แวร์
  - ค. การเขียนรายงานให้ลูกค้า
  - ง. การดูแลซอฟต์แวร์หลังการขาย
  - จ. การวิเคราะห์ไปราคา
39. ข้อใดตั้งราคาซอฟต์แวร์ได้เหมาะสมที่สุด
- ก. สมมติตั้งราคาโดยดูจากราคาในท้องตลาด
  - ข. บรรณารตั้งราคาโดยปรึกษาจากคู่แข่งไม่ได
  - ค. อภิสัทตั้งราคาโดยนำวิธีหลายวิธีมากำหนดแล้วเปรียบเทียบกับกำลังจ่ายของลูกค้า
  - ง. ทักซิณตั้งราคาโดยดูจากจำนวนบรรทัดของซอฟต์แวร์
  - จ. สุรยุพท์ตั้งราคาจากขนาดของซอฟต์แวร์
40. ข้อใดเป็นวัตถุประสงค์ในการทำ Pert Chart
- ก. ควบคุมการทำงานตามแผนที่วางไว้ ไม่ให้ล่าช้า
  - ข. ให้ทรัพยากรต่างให้เกิดประโยชน์เต็มที่
  - ค. แสดงให้เห็นว่ากิจกรรม แต่ละกิจกรรมควรเริ่มเมื่อใดและเสร็จเมื่อใด
  - ง. บริหารโครงการ ดูว่างานใดสามารถเร่งได้บ้าง
  - จ. ถูกทุกข้อ

ชื่อ \_\_\_\_\_

รหัสนักศึกษา \_\_\_\_\_

**คำตอบตอนที่ 1 (40 คะแนน)**

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
ก																				
ข																				
ค																				
ง																				
จ																				

	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
ก																				
ข																				
ค																				
ง																				
จ																				

**ตอนที่ 2 จงตอบคำถามต่อไปนี้ (20 คะแนน)**

1. จงบอกความเสี่ยงที่อาจเกิดขึ้นในการพัฒนาซอฟต์แวร์มา 6 ความเสี่ยง (6 คะแนน)

.....

.....

.....

.....

.....

.....

2. จงบอกความแตกต่างของ Dynamic verification และ Static verification (2 คะแนน)

.....

.....

.....

.....

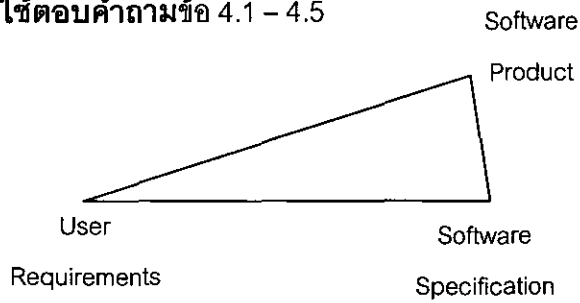
3. จงบอกความหมายของ test case ( 2 คะแนน)

.....

.....

.....

4. จากรูปจงใช้ตอบคำถามข้อ 4.1 – 4.5



4.1. จงบอกความสัมพันธ์ระหว่าง User Requirements และ Software Specification ที่เกิดขึ้นในภาพ (1 คะแนน)

.....

.....

4.2. จงบอกความสัมพันธ์ระหว่าง Software Specification และ Software Product ที่เกิดขึ้นในภาพ (1 คะแนน)

.....

.....

4.3. จงบอกความสัมพันธ์ระหว่าง Software Product และ User Requirements ที่เกิดขึ้นในภาพ (1 คะแนน)

.....

.....

4.4. การตรวจสอบความถูกต้องระหว่าง Software Specification และ Software Product เรียกว่าอะไร (1 คะแนน)

.....

4.5. การตรวจสอบความถูกต้องระหว่าง Software Product และ User Requirements เรียกว่าอะไร (1 คะแนน)

.....

รหัสนักศึกษา \_\_\_\_\_

5. ในการออกแบบซอฟต์แวร์ที่ดีควรมี Cohesion และ Coupling เป็นอย่างไร ( 2 คะแนน)

.....

6. จงวาดรูป Pert Chart ที่สอดคล้องกับตารางงานต่อไปนี้ (3 คะแนน)

งาน	งานที่ต้องเสร็จก่อน	เวลา (วัน)
A	-	2
B	-	3
C	A	1
D	A, B	5

รหัสนักศึกษา \_\_\_\_\_