

King Mongkut's University of Technology Thonburi  
Computer Engineering Department  
CPE 331 Software Engineering (Sections C and D)

Wongwattana 2 2 2550

March 6, 2008 9.00 – 12.00 hours

Student Name \_\_\_\_\_ Student ID \_\_\_\_\_

Instructions:

1. This final examination consists of 7 pages, plus this cover page. Do all your work in these examination sheets. Please **do not** use a separate booklet.
2. Read each question carefully. Follow instructions exactly. You will lose credit if you do not follow instructions.
3. For multiple choice questions, **circle** the letter of the correct answer (a through e). For other questions, write your answer in the space provided.
4. The amount of credit is indicated separately for each question. Total possible credit is 100 points.
5. Do not spend too long on any one question. If you get stuck on a question, skip it and return to it later.
6. You may use a *hardcopy or electronic* English-Thai dictionary for this examination. All sound must be turned off for electronic dictionaries. No other books or notes are permitted in the examination area.

Instructor

\_\_\_\_\_  
(Mr. Kurt T. Rudahl)

All questions are approved by the department chair.

\_\_\_\_\_  
(Dr. Suthep Madarasmi)

- .. Read the description of the following **class hierarchy**. Then, in the space below, draw a **UML Class Diagram** that correctly shows the classes in this hierarchy, including their properties and methods, and their relationships.

In a Paint program, there is a general class called **Shape**. Every shape has the properties **xPosition** and **yPosition**, two integer values which represent the location of the shape on the screen, and **color**, which is a string. **Shape** has the methods **move()**, which takes a new X and Y location to which the shape should be moved and **erase()**. It also has a method that each subclass will override, called **draw()**

**Shape** has three subclasses, **Triangle**, **Circle** and **Rectangle**. Each subclass has its own version of the **draw()** method, because it requires different actions to draw each one.

**Circle** has a property **radius**, an integer which is the radius of the circle. The center of the circle is the **xPosition** and **yPosition**.

**Triangle** has properties **x1**, **y1**, **x2**, **y2**, **x3**, and **y3**, which are the X and Y coordinates of the three corners of the triangle.

**Rectangle** has properties **width** and **height**. The **xPosition** and **yPosition** are assumed to be the coordinates of the upper left corner of the rectangle.

Draw your class diagram below (8 points):

2. In the class hierarchy described in Question 1, are there any methods that the **Triangle** class inherits from the **Shape** class? If yes, list the name of one inherited method below. (4 points)

- . In the class hierarchy described in Question 1, each subclass overrides the **draw()** method of the **Shape** superclass, because different operations are needed to draw a circle, a triangle or a rectangle. What is technical term for this situation? (4 points)
- Collaboration
  - Robustness
  - Polymorphism
  - Architecture
  - None of the above
1. What is the difference between **black box testing** and **white box testing**? (4 points)
- Black box testing is focused on the user interface only, while white box testing covers the whole system.
  - Black box testing assumes that you do not know anything about the implementation of a system, while white box testing uses implementation knowledge.
  - Black box testing uses legal input data, while white box testing checks the response to invalid or illegal input data.
  - Black box testing is manual, while white box testing uses automated testing tools.
  - None of the above.
5. What is CMMI (Cabability Maturity Model Integration)? (4 points)
- One of the diagrams included in the UML standard
  - A popular software development process paradigm
  - A tool conveniently for browsing, editing, compiling and debugging software systems
  - A framework to guide continous process improvement within a company or organization.
  - None of the above
6. Which of the following is a true statement about the **Extreme Programming (XP)** software development process? (4 points)
- It requires programmers to always work in pairs while developing software.
  - It defines many different artifacts that should be created as part of the development cycle.
  - It is a process paradigm from the 1980's which is rarely used today.
  - It was developed in a university environment and is based on formal software engineering theory.
  - None of the above.
7. Name one common type or category of **high level system architecture**. (4 points)
8. Give one example of a **software configuration management tool**. (4 points)

9. What is **stubbing**? (4 points)

- a. A design strategy that involves building "throwaway" programs that illustrate or explore certain aspects of a system design
- b. Creating a written test plan before starting system validation
- c. One of the key requirements of ISO 9001 certification
- d. Working with stakeholders to define Use Cases
- e. None of the above

10. Which of the following issues is most closely related to **deployment** of a software system? (4 points)

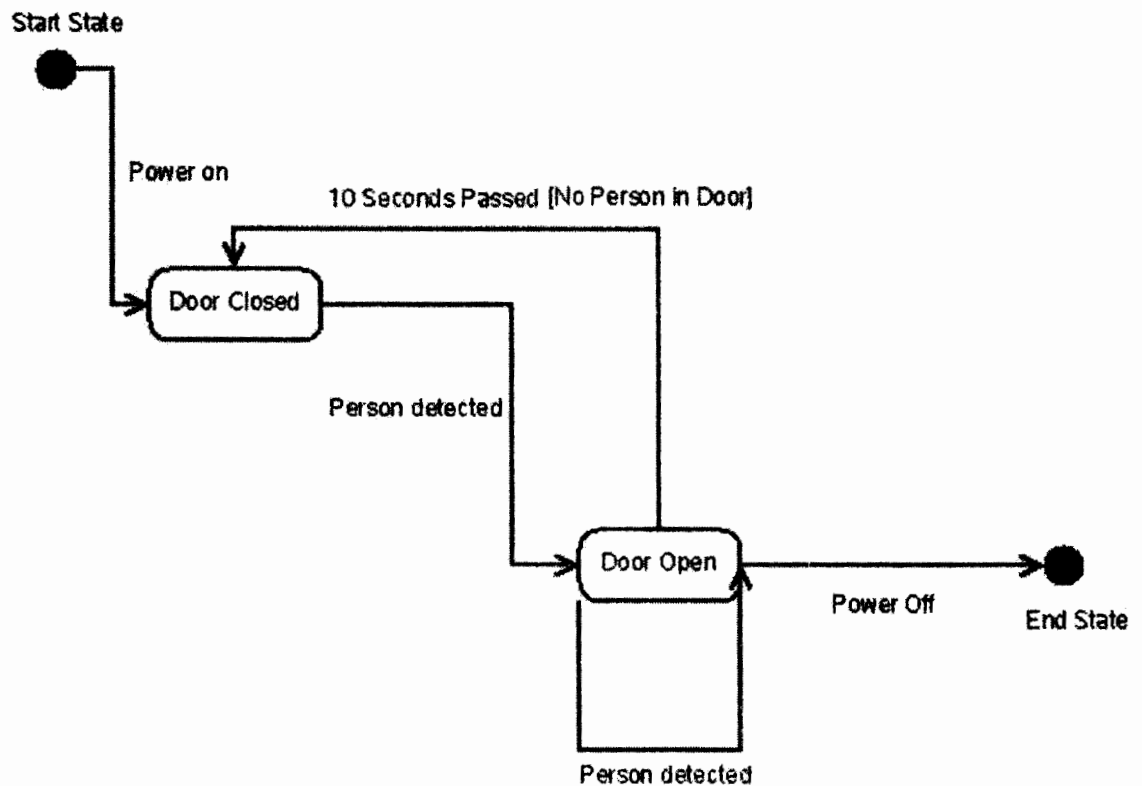
- a. Layout of user interface screens
- b. Choice of programming language and platform
- c. Communication protocols
- d. Migrating databases from legacy systems
- e. Setting up a source code repository

11. What is **refactoring**? (4 points)

- a. A strategy for software evolution that involves changing the structure of code while keeping the same functionality
- b. A design technique that starts with a general description of the system and then adds more detail
- c. The process of reusing code from one software system in a different software system
- d. A common method for validating software
- e. None of the above

12. Study the diagram below, then answer the questions that follow the diagram.

# Automatic Door



- What type of diagram is this? (4 points)
- There are two main types of graphical objects in this type of diagram. What are they called? (4 points)
- According to this diagram, what happens if the door is closed and a person is detected standing in front of the door? (4 points)
- According to this diagram, what happens if the door is open for ten seconds? In what situation could something different happen? (4 points)

13. Which of the following is the best example of a CASE tool? (4 points)
- a. make
  - b. Eclipse
  - c. Firefox
  - d. gcc
  - e. EditPlus
14. Suppose that you were going to design and build a new word processing software package. What would be the **name** of one important Use Case? (4 points)
15. What are **coding standards**? (4 points)
- a. The full set of diagrams defined by UML
  - b. The rules for creating a "make" file to automate the build of a software system
  - c. A set of rules that specify in detail how the programmers in a team should write their code.
  - d. The detailed list of requirements and constraints for a software system.
  - e. None of the above
16. Suppose that you have written a function called **checkDate**. This function takes one argument which is a character string, such as "19/02/2008" (dd/mm/yyyy) and checks whether the form of the date is correct or not. If the date is valid (legal and correctly formatted), then the function returns the value one, otherwise the function returns the value zero.
- a. Using C language syntax, write a **declaration** (also known as a **prototype**) for this function. (Hint: the date argument should be declared as **char \* date**.) (4 points)
  - b. Suppose that you were testing this function. What value should the function return for the test case "14/12/1999"? (4 points)
  - c. What value should the function return for the test case "14-DEC-1999"? (4 points)
  - d. In the space below, write a string (another test case) that you could use to test for another condition that **checkDate** should also handle, different from the conditions above. (4 points)

17. Which of the following concepts is most closely related to **Model Driven Architecture**? (4 points)
- a. Software configuration management
  - b. Code generation
  - c. Prototyping
  - d. Deployment
  - e. Linkable libraries
18. What high level system architecture did you use for your term project? List the name of the project and then the type of architecture, below. (4 points)