

OLLSCOIL NA hÉIREANN
The National University of Ireland, Galway

SEMESTER II
SUMMER EXAMINATIONS 1998/99

B.Sc. Degree Examination in Information Technology

SOFTWARE ENGINEERING III (CT417)

Professor D. Bell
Dr. G. Lyons
Mr. D. Chambers
Dr. S. Flynn

Time Allowed : Three Hours

Candidates should attempt five questions, at least two from each section.
Please use separate answer books for each section.
All questions carry equal marks

SECTION A

1. (a) Write a paragraph explaining what you understand by the term "software quality". (8)

(b) "The quality of a software product is completely independent of the quality of the process used to create it."
Argue for or against this statement. (12)
2. (a) Describe, using a diagram, the process of continuous improvement. (6)

(b) What is the Personal Software Process? Describe how it might lead to improved software quality. (14)
3. (a) You are given a program which, you are told, will take a positive integer and display the corresponding ordinal (1st, 2nd, 3rd, etc). Write test cases for this program.
What determines a successful test? (8)

(b) Describe three methods used for integration (incremental) testing. (6)

(c) Economics dictate that testing must eventually terminate. Describe useful stopping criteria. (6)

4. (a) Give definitions of (i) defect density and (ii) process yield. Explain how each measurement is used. (6)
- (b) Write a note on design defects and explain why these can be the most difficult defects to remove. (7)
- (c) Based on your own experience with the PSP, what steps would you take to prevent defects in your code. (7)

SECTION B

5. Identify possible objects in the following system and develop an object oriented design for them. You may make any reasonable assumptions about the system when deriving the design.

"A group diary and time management system is intended to support the timetabling of meetings and appointments across a group of co-workers. When an appointment is to be made which involves a number of people, the system finds a common slot in each of their diaries and arranges the meeting location and appointment for that time. If no common time slots or locations are available, it interacts with the user to rearrange their personal diary to make room for the appointment. The system also allows a privileged user category called "Group Manager" to maintain the system i.e. to modify the list of locations available and edit user contact details if necessary. Only a "Group Manager" has full access to modify / delete the contact details or bookings for any user. The system should not allow double bookings of locations or bookings that cover part of the same time period. Bookings can be made on a once off basis or as a repeating sequence e.g. the same time every week for 12 weeks. The system is client / server in nature and can be accessed through the Internet."

- (a) Derive a top level Class Diagram for this system, showing important attributes, methods and links between classes.
- (b) Should any of the resulting classes be defined as abstract classes or interfaces?
- (c) Discuss how the Proxy Design Pattern might be used in the design or extension of this system.

6. (a) Discuss the design guidelines provided in the Law of Dementor. Give an example of a design which breaks this law.

(b) Consider the following narrative:

"Students must take 6 modules to complete their course. Each module is examined and the students are awarded a final grade for the module. Each module is given by one or more lecturers. Each student is also assigned a lecturer to act as a director of studies for that student for the duration of their course."

(i) Draw a UML use case diagram for this system.

(ii) Derive a class diagram for the system.

(iii) What would happen if the same student, for some reason, takes the same module twice? How would your design cope with this situation?

7. (a) Assume that we have a class List available and that we wish to implement a class AddressBook in which the addresses are stored in a List. How would you model the association between List and AddressBook? Explain your answer in terms of best design practice. How should clients of the AddressBook class interact with it to add or remove addresses from the List?

(b) Using UML style graphical notation, design the following objects. Use your own experience to decide on the attributes and operations associated with these objects:

(a) A telephone.

(b) A printer for a personal computer.

(c) A personal stereo system.

(d) A bank account.

(e) A library catalogue.

Based on these designs, write precise interface definitions in JAVA or C++ for each of these object classes.

8. (a) Explain what is meant by the Liskov Substitution Principle and give a suitable example of its use.
- (b) Describe what is wrong with the following code sample (in terms of good design practice). Suggest a better design approach based on using the dependency inversion principle.

```
void paint() // Method within some container class
{
    while (shape.next() != null)
    {
        switch(shape.type)
        {
            case: Shape.CIRCLE:
                drawCircle((Circle) shape);
                break;
            case: Shape.RECTANGLE:
                drawRectangle((Rectangle) shape);
                break;
        }
    }
}
```

- (c) Write the classes / interfaces for the observer design pattern in JAVA. Add operations for attaching new observers to a particular publisher. Is it necessary for publishers to “know about” all the attached observers ? Consider what would happen if the observers and publishers were distributed i.e. they exist in different processes.