THIS PAPER IS NOT TO BE REMOVED FROM THE EXAMINATION HALLS

UNIVERSITY OF LONDON **291 0210**

BSc Examination
for External Students

COMPUTING AND INFORMATION SYSTEMS

Software Engineering and Development

Dateline: Monday 18 May 2009 : 10.00 – 1.00 pm

Duration: 3 hours

This paper consists of six questions. Candidates should answer **FOUR** questions. Full marks will be awarded for complete answers to **FOUR** questions. Do not attempt more than **FOUR** questions on this paper.

A hand held calculator may be used when answering questions on this paper but it must not be pre-programmed or able to display graphics, texts or algebraic equations. The make and type of machine must be stated clearly on the front cover of the answer book.

Question 1.

a) Give a brief definition of each of five internal qualities of software projects **[10]**

b) Describe the waterfall model of software development. Your answer should include a description of each of the main stages, namely: analysis, design, coding, testing and project maintenance **[10]**

c) From your experience of software development or otherwise, comment on the accuracy and usefulness of the waterfall model **[5]**

Question 2.

a) Briefly describe the two classical software project size metrics. **[4]**

b) Name 3 advantages each metric has over the other **[6]**

c) Describe and give formulae for the four software process metrics—productivity, adequacy, unit cost, and documentation—that are computable using function points **[8]**

d) Give the formula for the function points metric size and illustrate how we can estimate a software project with it using as an example a project characterised by the following function point parameter values: **[7]**

Parameter	*count*	*weight*
User Inputs	5	4
Outputs	9	2
Inquiries	4	6
External Interfaces	3	3
Files	8	5

and function point topic influences:

F	*Topic*	I_i
1	Reliable back-up and recovery	3
2	Data communications	1
3	Distributed processing	0
4	Performance requirements	2
5	Shared environment	3
6	On-line data entry	3
7	Non-atomic data transactions	2
8	On-line master file update	1
9	Complex files and queries	0
10	Complex internal processing	0
11	Code designed for reuse	2
12	Conversion/installation included	4
13	Many sites in many organisations	2
14	Promotion of change	3

Question 3.

a) Briefly describe each the Critical Path Method and the PERT technique in terms of their use in constructing a software project schedule. **[6]**

b) List and explain three kinds of information that are useful for developing the software project schedule and can be inferred by a combination of the two methods. **[9]**

c) Consider the task of developing software for a Library Information System (LIS) for tracking and managing library resources. The software library info system should: display the library catalogue, provide search options, and periodically send reminders for overdue resources. There are significant differences between faculty and student users.

The scheduling of this task must account for the following requirements (the subtasks are given in italic):

- initially there has to be designed a central object *library user* (T1) class taking no more than a working day;
- next, the classes for *faculty* (T2) and *students* (T3) are designed, each taking two days, in parallel with the design of
- the *library staff management module* (T5) which takes three days but it should be preceded by a subtask for *overall control* (T4) which requires a day, so the total for these two is four days;
- after all these parallel subtasks, a database called *library catalogue* is created (T6) taking two days. It maintains information about the library resources and their catalogue signatures and must be completed before the following work begins;
- the design of the *overall control* (T7) then takes up to seven days as it has to establish communication between the catalogue database and its users that have to be ready at the final seventh day:
- a class is designed for *providing faculty access to the system* (T8) which takes five days,
- and one for *student access to the system* (T9) which takes three days;
- along with the control design the *software interface* (T10) needs to be designed, taking up to four days;
- the process terminates with *testing* (T11) and *quality assurance* (T12) conducted taking two days each, in succession.

Produce a Gantt Chart for scheduling the development of this software LIS system, including a milestone before each subtask in the main working sequence having a common one for the parallel subtasks **[10]**

Question 4.

a. What is software maintenance and why is it such an important part of the software lifecycle? **[5]**

b. What are alpha and beta testing? **[4]**

c. Suppose that an error is found at a beta test site. How can program slicing be used as an aid to determining the cause of the fault? **[6]**

d. Define what is meant by a *static backward* slice of a program What line numbers are included in the thinnest static slice of the program below for the slicing criterion ({x},9)? **[10]**

```
BEGIN
1:      x := 0;
2:      y := y + 5;
3:      x := x + y;
4:      y := y + 1;
5:      IF x > 9
6:         THEN y := y * y;
7:         ELSE  y := y * y + 1;
8:      i := y;
9:      WHILE i < 10              DO
10:             i  := i +1;
11:             x := x *i;
12:     ENDWHILE
13:     y := 0;
14:     END
```

Question 5.

a) What are the main components of a Booch Class Diagram? **[5]**

b) You are asked to produce a computer-based system for keeping track of courses for a university department. The system has the following characteristics:

All Professors are lecturers; Professors design courses and lecturers teach courses.
There are up to 30 lectures and up to 10 seminars in a course
A course is assessed by one exam and up to 4 courseworks.
A student takes up to 6 courses
A program consists of 12 courses.

Draw a Booch Class Diagram for the system

[20]

Question 6

a) What is the 7 ± 2 rule and what is its relevance to computer interface design? **[5]**

b) Contrast the human mind's abilities at Recognition and Recall, and discuss the relevance of this to interface design. **[5]**

c) What does it mean to say that human memory is associative? Recount an experiment that suggests that memory is associative. What implication has this for interface design? **[5]**

d) From your reading, or from your personal experience as a software developer or a computer user, name five features of interfaces that are either particularly helpful or particularly disruptive. Explain your answers. **[10]**

END OF PAPER