UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2000

MEng Honours Degree in Information Systems Engineering Part IV
MEng Honours Degree in Mathematics and Computer Science Part IV
MEng Honours Degrees in Computing Part IV
MSc in Advanced Computing
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the Associateship of the City and Guilds of London Institute
This paper is also taken for the relevant examinations for the Associateship of the Royal College of Science

PAPER C475=I4.16

SOFTWARE ENGINEERING - ENVIRONMENTS

Friday 19 May 2000, 10:00
Duration: 120 minutes

Answer THREE questions

Paper contains 4 questions

Section A *(Use a separate answer book for this Section)*

1 Many people in our industry are always ready to jump on the next bandwagon that promises improvement to the quality of software developed. One of today's bandwagons is *extreme programming* (XP).

a Pair programming is one of the novel techniques of XP.

i) What is pair programming?

ii) What are the benefits that pair programming is supposed to provide?

iii) What disadvantages do you think there are with pair programming?

b Refactoring is another novel technique of XP.

i) What is refactoring?

ii) What are the benefits that refactoring is supposed to provide?

iii) What disadvantages do you think there are with refactoring?

c Consider a place of programming employment with which you are familiar. How close was the methodology used to the XP ethos? What changes would have to be made to use XP in this environment? Do you think if XP had been used, there would have been an improvement or deterioration of the resultant product, the work environment and customer satisfaction? Justify your answers.

If you have never worked as a programmer, consider any academic group programming work.

d Extreme programming can be considered to consist of a collection of different techniques. Which of these techniques would you *least* like to see in the next programming process fad. Justify your answer.

The four parts carry, respectively, 25%, 25%, 40% and 10% of the marks.

2 Every programming language has its own peculiarities that can make programs difficult to read and maintain. Teasing these out and changing to more readable constructs makes programs more maintainable.

a For *one* of the following languages: C, C++, FORTRAN, Perl or Python, choose two constructs which can easily obscure code. Show how in your chosen language, the constructs could be replaced to produce more readable code. Are there disadvantages in using your replacements?

b The following is a snippet of code that accepts filenames with a `.j` suffix. No design documentation exists. The maintenance task that needs to be undertaken is to rewrite it so it now accepts code with a `.jav` suffix.

```
char *c_name(char *s){
    char *b, *s0;
    int c;
    b=s0=s;
    while(c=*s++)
        if (c=='/') b=s;
    if (--s<s0+3 || s[-2]!='.'
                 || ((c=*--s)!='j')) {
        infname=s0;
        Fatal("filename must end in .j");
        }
    *s=c;
    return b;
    }
```

i) Rewrite the conditional to meet the new specification. You should make sure your code is more readable than the original.

ii) The code above actually accepts files that it shouldn't. What is the format of the illegally named filenames? Does your answer to part b i of this question accept similar files? If it does, how would you change it so that it no longer accepts illegal program file names?

The two parts each carry 50% of the marks.

Section B *(Use a separate answer book for this Section)*

3a Briefly outline the engineering and economic arguments to support the need for requirements engineering?

b What is functional decomposition? Briefly explain why it is often unsuitable for dealing with the development of large, complex systems.

c i) According to Michael Jackson, what are problem frames and why are they useful in software development. Give one example of a problem frame to illustrate your answer.

ii) Distinguish indicative descriptions from optative descriptions, and give an example of each kind.

d i) In developing software for your individual project at Imperial College or elsewhere, comment on the maturity of your development process, and briefly discuss ways in which you can improve your process maturity. Use the CMU/SEI's Capability Maturity Model to explain your answer.

ii) At what level of process maturity is it appropriate to introduce requirements engineering. Briefly explain your answer.

The four parts carry, respectively, 15%, 15%, 30%, 40% of the marks.

4a Explain briefly the role of inspection in requirements engineering, and provide an outline of an inspection process of a requirements document.

b Outline a requirements elicitation process for a web-based system to collect and process student feedback on taught courses at Imperial College. State and briefly justify:

i) the stakeholders for which requirements have to be elicited,

ii) the type of information you want to collect, and,

iii) the technique you would use to elicit requirements in this particular case.

c Briefly describe a technique for validating the requirements elicited in part (b).

The three parts carry, respectively, 25%, 60%, 15% of the marks.