THE BRITISH COMPUTER SOCIETY

THE BCS PROFESSIONAL EXAMINATION
Certificate

SOFTWARE DEVELOPMENT

19th October 2001, 2.30 p.m.-4.30 p.m.
Time: 2 hours

SECTION A

Answer TWO questions out of FOUR. All questions carry equal marks.

*The marks given in brackets are **indicative** of the weight given to each part of the question.*

1. *a)* Develop pseudocode to process student examination marks from the file '*rawmarks*' thus:

Each student record contains a name (20 characters) and an identifier (6 digits) followed by six integer marks expressed as percentages. For each student add to his/her record 'PASS', 'RESIT' or 'FAIL' according to the following criteria:

'PASS' : All six examination marks are passes.
'RESIT': Either 5 marks are passes and one mark is below 40% or 4 marks are passes with two marks lying between 35% and 40%.
'FAIL' : All other cases.

An individual mark is deemed a 'pass' if it is 40 or more percent.

In addition to the processing outlined above, create two output files:
'passlist' containing the names and identifiers of only those candidates who have passed
'outfile' a printer/character file containing the students' names, identifiers, exam marks, average mark and overall results.

Separately count the number of pass, fail and resit candidates and append this information to the end of *'outfile'* with suitable captions.

You must show adequate development and give at least two stages of pseudocode development. Actual program code is not required. **(24 marks)**

b) Describe suitable data structures to use with the program in a programming language of your choice. State which language you are using. **(6 marks)**

2. The following program is intended to put the contents of three integer variables into descending order:

```
Line no.    code
 0          PROGRAM order
 1          INTEGER VARIABLES num1, num2, num3
 2              PROCEDURE whatdo (INTEGER VARIABLES x, y)
 3              LOCAL INTEGER VARIABLES  temp
 4              temp ← x
 5              x    ← y
 6              y    ← temp
 7              END whatdo
 8          BEGIN  {code for 'order' - top level}
 9          READ (num1, num2, num3)
10          IF num1 IS LESS THAN num2 THEN CALL whatdo (num1, num2)
11          IF num2 IS LESS THAN num3 THEN CALL whatdo (num2, num3)
12          WRITE (num1, num2, num3)
13          STOP
```

a) Dry run the above code with input values 1, 2 and 3. **(12 marks)**

b) What action is performed by the PROCEDURE whatdo? **(2 marks)**

c) Modify the given code so that the stated purpose of the program is achieved with <u>any</u> three input values. Clearly indicate where the modified code is placed in the supplied code, which need not be written out in full. **(10 marks)**

d) Specify a set of test data values sufficient to fully test the code. **(6 marks)**

3. *a)* Specify and discuss the principles of multiple module program construction. **(24 marks)**

b) Describe the part that the client should play to carry out the task successfully. **(6 marks)**

4. *a)* *Structured Programming* is a technique to reduce errors when programming. With respect to algorithm design, Structured Programming defines certain rules for the construction of algorithms. Describe at least THREE of these rules of Structured Programming, as they might be expressed for a modern programming language[*1]. **(10 marks)**

b) *Defensive programming* is a technique you can use to improve your program's ease of maintenance and robustness in use. Describe at least THREE techniques of Defensive Programming, as they might be expressed for a modern programming language. **(10 marks)**

c) Compare and contrast the terms *Structured Programming* and *Defensive Programming* with reference to any modern programming language with which you are familiar. **(10 marks)**

[*1] a 'modern' language may be considered as any third-generation or visual programming language.

NOW PLEASE ANSWER QUESTIONS FROM SECTION B OVERLEAF ➜

SECTION B

Answer FIVE questions out of EIGHT. All questions carry equal marks

*The marks given in brackets are **indicative** of the weight given to each part of the question.*

5. The number of combinations (C) from M items taken N at a time is given by

C = fac(M) / [fac(N) * fac(M - N)]

where fac(X) = X * (X - 1) * (X - 2) * … * 2 *1

Write a function to evaluate fac(X) and incorporate it in a program to request input values of M and N and calculate C. Provide appropriate captions and state which language you are using. **(12 marks)**

6. An old computer program which processes car mileage and cost data accepts ONE pair of input data items per week. These are the amount of money spent (pounds sterling) and the price in pence per litre of the petrol bought. The program cannot accept more than one pair of data items for any one week and cannot be amended. However in a week several petrol purchases may be made at different petrol prices per litre, not foreseen by the original programmers.

Develop the logic and write pseudocode which converts several such pairs of data items to an equivalent single transaction with an appropriate petrol price in pence per litre, so that the same volume of petrol would have been purchased. **(12 marks)**

7. Describe how to implement BOTH a stack AND a queue using

a) arrays **(6 marks)**

b) pointers **(6 marks)**

8. Describe the operation of a modern file/database access mechanism such as VSAM or ISAM. Include a description of deletion, updating and insertion of data, as well as a commentary on its strengths and weaknesses. **(12 marks)**

9. You have been asked to plan the development of a website for a local small business. Describe the documentation you would specify as part of the project's deliverables. Give your reasons for the specified documentation and include any assumptions you make about the needs of the small business. **(12 marks)**

10. Discuss the impact of software tools (such as web page generators, and application generators) on the testing plan of a small to medium sized commercial software project. Comment especially on the relative merits of black box and white box testing, the testing plan (whether top-down or bottom-up) and any other verification and validation activities that you consider important. Give your reasons. **(12 marks)**

11. Describe with suitable examples, what the term *abstract data type* means. Your answer should consider at least THREE different examples with illustrations of their use. **(12 marks)**

12. Computer systems are often described using the metaphor of a set of concentric (onion-like) rings. State whether this is a useful idea and describe one computer system in this way to justify your answer. **(12 marks)**