

THE BRITISH COMPUTER SOCIETY

THE BCS PROFESSIONAL EXAMINATION Diploma

SOFTWARE DEVELOPMENT ENVIRONMENTS

30th April 2002, 2.30 p.m.-4.30 p.m.

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

Time: TWO hours.

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

1.
 - a) Name a software development method of your choice and identify its stages. **(4 marks)**
 - b) Pick THREE stages from your chosen method and, for each stage, name an appropriate language or notation and explain what contribution the stage makes to the overall software development. **(9 marks)**
 - c) Suppose you were a programmer given the task of making a significant change to a piece of software. You have not yet seen the documentation for the software. List FOUR kinds of information that you would hope to find in the documentation and state why they are needed and how they would be used. **(12 marks)**

2.
 - a) Describe FIVE principle features common to modern operating systems. **(10 marks)**
 - b) Name TWO operating systems known to you and describe TWO differences between them that are not based solely on the user interface. **(5 marks)**
 - c) The two extremes of operating system interfaces are command-line interface and WIMP (Window, Icon, Menu, Pointer). Briefly describe how each interface is used and give ONE advantage for each type. **(10 marks)**

3.
 - a) When a program is executing, data can be read from a file and results can be written to a file. Describe and compare the following terms relating to files:
 - i) Text file and binary file
 - ii) Sequential and direct access
 - iii) Local and remote
 - iv) Open for reading, writing and appending**(18 marks)**
 - b) In most programming languages there is a file *open* and a file *close* command. Describe the purpose of these two commands and include information on the parameters which are used. **(7 marks)**

4. a) Program editors within program development environments often provide features that aim to minimise the introduction of errors into a program. Describe THREE such features and explain how each reduces the possibility of errors. **(9 marks)**
- b) Describe FIVE features of a debugging tool that enable it to assist the process of identifying the source of errors in a program. **(10 marks)**
- c) In order for a debugging tool to operate successfully, additional code has to be inserted by the compiler. The inclusion of this additional code is achieved by setting the appropriate compiler option. Identify and explain the purpose of THREE other compiler options that a program development environment may offer. **(6 marks)**

5. a) Compare and contrast two possible approaches to the identification of test cases that could be used to adequately test computer software. You should clearly identify the advantages and disadvantages of each approach. **(12 marks)**
- b) Consider the pseudo code below and clearly identify the test cases that should be used to ensure the correct operation of the program.

```
Procedure leapyear (year)
BEGIN
    If (year > 1582)
        If (year is divisible by 400) Or
            (year is divisible by 4 but not 100) Then
            Display "Is a leapyear"
        Else
            Display "Not a leap year"
    Else
        Display "Unable to decide"
END leapyear
```

(5 marks)

- c) Describe in detail a software tool that can be used to assist the testing process. Your answer should highlight the benefits of using the tool, give an example of where it would be appropriate to use it and identify any issues with which a user of the tool should be familiar. **(8 marks)**
6. a) Software quality is often expressed in terms of quality factors such as *re-usability*, *portability*, *maintainability*, *reliability* and *clarity*. In the context of software engineering, explain the terms in italics and indicate how each may be achieved. **(10 marks)**
- b) Having observed several different commenting styles and standards in the code produced by different software engineers in a large software house, the quality division has decided that programming style should be standardised. Write a document that defines a suitable standard. You may assume the company uses a programming language with which you are familiar. **(15 marks)**