THE BCS PROFESSIONAL EXAMINATION
Certificate

APRIL 2002

EXAMINER'S REPORT

SOFTWARE DEVELOPMENT

General Comments

A poor selection of questions contributed to some students' downfall. Candidates must recognise just what they can and cannot do at the outset. Moreover, there is no time to write out neat copies of answers first written in rough - it is often sensible to jot notes down in rough initially but not to re-write full answers. Those who did this invariably answered too few questions. Too many candidates again ignored the rubric and did three A-section questions or six B-section ones. Merely copying out the questions into the exam book will gain no marks at all.

There are many methods available for stepwise programme development. Generally the use of flowcharts is inappropriate but was not penalised. Candidates should compare the time taken to draw a flowchart, especially with ruled lines, against written pseudocode.

QUESTION ONE

1. A western-style name is held in an array of characters called 'namestr'; the full name is delimited by quotes thus:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
'	P	A	U	L		J	O	H	N		W	O	O	D	S	'

Forenames **Last name**

The name is required to be output in the following format:

<Last name in full>, < Initial of first forename >. < Initial of second forename >. <……>.

For example, the name

'PAUL JOHN WOODS'

is required to be printed as

WOODS,P.J.

Develop an algorithm for this process and show adequate development so that subsequent coding is straightforward. State your target language.

The data has already been validated so you are NOT to deal with error situations: thus there will always be *at least* one forename, the space

character will be present between the names and the array is big enough to hold all the characters. **(30 marks)**

An unpopular and poorly attempted question. A significant number of candidates just read in the example 'JOHN PAUL WOODS' and printed the appropriate parts from that. Some marks were given if they showed how to use an array subscript, but this was a serious misunderstanding of the question.

Those who struggled with this problem and wrote highly complex and involved answers will be surprised at how elegant and simple the answer really is. (See below.) The answer hinges on manipulating the subscript of the array 'namestr'; any that did not realise this struggled. The usual reason for inadequacy was that nearly all failed to write an appropriate first algorithm or even tried to write (usually Pascal) code straight away. Even if some stages are obvious, it is best to start with an algorithm that looks specifically at how to solve the particular problem and not wrestle immediately with the nitty gritty of language detail. Far fewer were those who had too many development stages, and never really got anywhere. Unfortunately it is a false saving here to try to cut out early stages, assuming they are obvious, and descend to code immediately.

Answer Pointers

Remember - the question said the data had been validated so there is no need to incorporate checks on the format.

1st stage

find last name → start (first)
find last name → finish (last)
write last name
output ','
From the front of the array
WHILE NOT start of last name DO
write initial of forename
output '.'
skip the rest of characters in forename and the following space
END WHILE

To find where the last name starts and ends in the array 'namestr' it is useful to realise we can start from the right-hand end of the name, marked by the last quote stored. The subscript can then be decremented until the first space character. Thus in the example

……'namestr' subscript (sub) values
……

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
'	P	A	U	L		J	O	H	N		W	O	O	D	S	'

↑ ↑

← Forenames → space before Lastname
last quote

(It is essential to realise that other names like ‘FREDDY MAITLAND’ will have the desired parts in different parts of the array.)

We can {find last name} by

skip first ‘quote’ character
    using sub as an index
        skip characters until last ‘quote’ character found sub → last
        decrease ‘sub’ until ‘space’ character encountered sub + 1 → first

The last name can now be output by

    FOR sub := first TO last DO WRITE namestr[sub]

Looking again at the example

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
‘	P	A	U	L		J	O	H	N		W	O	O	D	S	‘

↑ ↑ ↑ ↑

first initial subsequent initials ‘first’ (found above) ‘last’

ALWAYS at 2nd position ALWAYS preceded by ‘space’ character

We can now develop {write initial of forename} thus:

```
sub := 2
WHILE  sub < first  DO
        WRITE (namestr[sub]);
        WRITE (‘.’)
        WHILE namestr[sub] <> space DO sub := sub + 1
        increase sub by 1 to skip space
END WHILE
```

We are now in a position to code the developed algorithm. Language-specific features can now be used; here the target language is a PASCAL like language.

```
{find last name}
        sub := 2
        WHILE namestr[sub] <> quote DO sub := sub + 1;
        last := sub - 1;          { position of LAST character of last name }
        WHILE namestr[sub]  <> space DO sub := sub - 1;
        first := sub + 1;                 { position of FIRST character of last name }

{write last name}
        FOR sub := first TO last DO WRITE (namestr[sub]);
        WRITE (‘,’);

{write initials of forenames}
        sub := 2;
        WHILE sub < first DO
           BEGIN
                WRITE ( namestr[sub]);        {write initial }
```

```
        WRITE ('.');
        WHILE namestr[sub]  <>  space DO sub := sub + 1;
        sub := sub + 1          {skip space}
    END
```

QUESTION TWO

2. *a)* Describe a *Software Development Kit (SDK) or* a *Software Development Environment (SDE)* with which you are familiar. Be sure to explain the purpose of the software that the kit or environment is to produce. (12 marks)

***b)* What generic elements or features should be present in a good SDK or SDE, and why? Be sure to link your answer to types of software development practice that you know about. (8 marks)**

***c)* What do you think is driving the changes in software development practice? How will these changes be reflected in new SDK or SDE products? Make your reasons plain. (10 marks)**

Answer Pointers

a) The description should address functionalities and applications of any SDK/SDE known to the candidate. This can range from environments to develop embedded microprocessors through to dynamic debug environments for production and test of 3GL and 4GL programming or any other descriptions that show a clear understanding of the specifics of a support environment for a particular software development process.

b) This part of the question asks the candidate to identify higher-level activities with the general development process. It expects the candidate to use the specific descriptions of part (a) above.

Functionalities might include such things as

- Test environments – set value, trace values, start and stop points
- Coding environments – on-screen syntax aids, libraries of features for inclusion and use
- Linking tools that create an executable module from part-compilations or part-assemblies
- Analysis tools that identify calling sequences and execution paths
- Pretty-print tools that structure code for readability and maintenance
- On-screen HCI type drag-click actions that mimic or model the build process e.g. MS-Access 'form design' screen

c) This is a 'differentiator' question, looking for answers that appreciate the craft nature of programming and the economic imperative to obtain more productivity, more accuracy, more 'productionising'. Craft programming is still evident in the

re-use (libraries) or auto-debug (testing harnesses) or productivity-aid (on-line syntax assistance).

Expected changes to implement 'productionising' are things like 'standard parts' and methods of binding standard parts for interchangeability, ideas of architecture styles such as patterns or the OO idea of classes and instances.

QUESTION THREE

3. A key and a data item are stored in pairs in the following array

integer array dataset(1:2;1:n)

such that any one pair in dataset might look like dataset(key-x, value-x). value-x is never zero so it can be used as a test value.

A search algorithm is proposed as follows:

```
integer function  Afind (input integer: dataval, range);
begin

comment dataset is a global two-dimensional array 1:2 by 1:range
comment This algorithm determines whether 'dataval' value is in dataset.
comment 'range' is the number of pairs in dataset.
comment If 'dataval' is found, the return value of Afind gives the associated
    key
comment If 'dataval' is not found, the return value of Afind is zero

local integer  key, pos;

key := 0;
pos := 0;
while key = 0 and range > pos do
begin
    pos := pos + 1;
    if dataval = dataset(2,pos) then key := dataset(1,pos)
end;

Afind := key;

end function  Afind;
```

a) **Given the following table of keys and values representing dataset:**

12	**4536**
17	**8723**
9	**1625**
16	**9165**
14	**1948**
18	**1984**
10	**7125**
11	**2638**
8	**5128**

15	**2584**

Dry-run the execution of Afind (9165, 10). What is the value returned by Afind? **(15 marks)**

b) **Describe the expected performance of the algorithm, making plain any assumptions you need to make.** **(5 marks)**

c) **What features of the data might significantly change your estimate of performance?** **(10 marks)**

Answer Pointers

Afind is a scanning algorithm, looking serially for the data until it finds it. The key in this case is the sum of the first three digits in the datavalue.

a) Dry run Afind (9165,10). !0 points available for clear sense of sequential searching.

Dataval	***Range***	***Pos***	***Dataset index= (2,pos)***	***Dataset(2,pos)***	***Key @(1,pos)***
9165	10	0	N/a	N/a	
		1	(2,1)	4536	
		2	(2,2)	8723	
		3	(2,3)	1625	
		4	(2,4)	9165	16

b) Expected performance is Big-O(n) meaning performance is of the order of the number of cells in the search space. This measure of performance is technology-independent, concentrating wholly on the algorithm.

Specific notation of Big-O is not necessary if clear sense that performance is directly dependent on the size of the space to be searched because the scanning is linear. Marks were awarded if performance is quoted with concept of (1/2n) indicating some kind of 'averaging' in the analysis of the candidate.

c) Marks here were for candidates who recognise the scanning algorithm and can address its known pitfall in their answers.

It is possible that candidates might offer features that improve performance. Where these answers do not imply or require a change of algorithm they will be evaluated sympathetically. Otherwise such answers will be marked at face value for the insight they show into the scan-search algorithm.

QUESTION FOUR

4. **A PC-based stock enquiry system is to be used in a works chemical store to facilitate enquiries for specific chemicals and to expedite stocktaking. Each chemical has a record structure, part of which is shown below:**

CHEMICAL.
NAMES.

Chemical-name.	**30 characters.**
Industrial-name.	**30 characters.**
Short-name.	**10 characters.**
Catalog-ref.	**10 characters.**
LOCATION.	
Row-number.	**1 character and 1 digit.**
Aisle-number.	**1 character and 1 digit.**
Shelf-number.	**1 digit.**
HAZARDS.	
Hazard-Number.	**1 digit.**
Classification.	**10 characters.**
Poison-Rating.	**10 characters.**

a) **Specify this record structure in an appropriate programming language. State clearly which language you are using.** **(6 marks)**

b) **Specify a file or table to hold up to 1,000 such records.** **(4 marks)**

c) **Write pseudo-code to handle a series of enquiries for different chemicals. An enquiry may use any of the entries under 'NAMES'. For those whose entries are found, the data stored under 'LOCATION' is to be printed. If an entry is not found, 'NOT AVAILABLE' is to be printed. If the 'Hazard-Number' is greater than 3, 'CHARTERED CHEMISTS ONLY' is also to be printed. If the entry in 'Poison-Rating' is 'SEVERE', 'RESTRICTED ACCESS' is also to be printed.**

After each enquiry the user is asked whether he/she wishes to terminate or continue with another enquiry. When the last enquiry has been processed, the number of successful and unsuccessful enquiries is to be displayed. **(20 marks)**

A quite popular and moderately well done question.

(a) Only a few reflected the given structure well in the code. Most had repeated character declarations, even for items which were obviously numeric. Use of short type names is satisfactory here as they do save time in this kind of question.

(b) Few realised what was wanted here. A declaration of the file/table structure <u>and</u> a brief indication of how to populate it were needed for full marks. If only two marks are available, students should realise that a page of code is not expected.

(c) This can be done with little development, so it was quite well done. Scope of long loops must be indicated. Earlier declarations need not be repeated here.

Answer Pointers

(a) An appropriate record structure for the given record structure in PASCAL :

```
TYPE        longstr  : PACKED ARRAY [1..30] OF CHAR;
       shortstr : PACKED ARRAY [1..10] OF CHAR;

       chemical = RECORD          {outer record}
```

```
names = RECORD                  {1st inner record}
            chemical_name,
            industrial _name : longstr;
            short_name,
            catalog_ref : shortstr
        END;
location = RECORD               {2nd inner record}
            row_number,
            aisle_number : ARRAY [1..2] OF CHAR;
            shelf_number : shortint
        END;
hazards = RECORD                {3rd inner record)
            hazard_number : shortint
            classification,
            poison_rating : shortstr
        END;
END {overall record structure}
```

(b) There were many possible answers to this depending on choice of language. Both a declaration of the file or table, and an indication of how to populate were expected. Thus in PASCAL:

```
datafile : FILE OF chemical;

WHILE NOT end_condition DO
    WITH chemical DO
        WITH names DO READ(chemical_name, industrial_name,
                           short_name, cataog_ref)
        END;
        WITH location DO READ (row_number, aisle_number, shelf_number)
        END;
        WITH hazards DO READ( hazard_number, classification,
poison_rating)
        END;
    END;
```

(c)

```
set counters to zero    (fail_ct = success_ct = 0)
REPEAT {for each enquiry}
    INPUT type of enquiry [1,2,3,4] → enq
    CASE enq OF
        1 : chemical name enquiry      .        SEARCH file matching
the
        2 : industrial name enquiry    .        input enquiry to
appropriate
        3 : short name enquiry         .        record field (match may
        4 : catalog enquiry            .        or may not be found)

    SET found = true/false depending on CASE
    IF found THEN
        BEGIN
            increment success_ct
            WRITELN (location details from chemical record)
```

```
            IF hazard_number > 3 THEN WRITELN ("chartered chemists only")
            IF poison_rating = "severe" THEN WRITELN ("restricted access")
        END
    ELSE  {not found}
        BEGIN
            increment fail_ct
            WRITELN ( "not available" )
        END
    WRITELN ("another enquiry?")
    INPUT [1 = yes  0 = no ] → ind
UNTIL  ind = 0
{program termination}
WRITELN (success_ct " successful enquiries;  "  fail_ct " failed enquiries" )
END.
```

QUESTION FIVE

5. Using an appropriate programming language:

a) **Define a data structure for one element of a linked list to contain a string of 3 characters as its data item.** **(2 marks)**

b) **Write code to read in and create a *linked list* of these elements where each new member is added to the head of the list. Input is terminated by '***'.** **(6 marks)**

c) **Write a function/procedure named 'length' which counts how many elements are present in the list.**

State the programming language you have used. **(4 marks)**

Answers divided almost completely into two categories:
a 'small group' who knew this material well and gained a high mark
a 'large group' who knew very little, especially pointer manipulation

The latter group usually wrote from memory on learnt examples, which usually did not answer the question. Others wrote about stacks and queues, 'pop' and push' functions which were not mentioned in the question.

Clearly answers coded in a stated language were expected here, languages are decidedly different concerning pointers.

Students should constantly ask themselves 'Is what I am writing answering the question?'
Irrelevant details from memorised programs do not impress the examiner.

Answer Pointers

(a)
```
TYPE  ptr = ↑node;
      node = RECORD
               data : string;  (or ARRAY [1..3] OF CHAR);
               next : ptr
             END;
```

(b)
```
VAR astring : string; p : ptr; head : ptr;
```

```
BEGIN
   WRITE ("list set up in inverse input order");
   head := NIL;
   READLN(astring);
   WHILE astring <> '***' DO
        BEGIN
              NEW(p);
              p↑.data := astring;
              p↑next := head;
              head := p;
              WRITELN ( "input next string… *** to finish");
              READLN( astring )
        END
END;
```

(c)
```
FUNCTION length  (place : ptr ) : INTEGER;
VAR  ct : INTEGER;

BEGIN
   ct := 0;
   IF place = NIL THEN WRITELN ('list empty')
   ELSE
      WHILE place <> NIL DO
      BEGIN
         ct := ct + 1;
         place := place↑.next;
      END;
   END  {IF }
   length := ct
END  {length function }
```

QUESTION SIX

6. Write a function 'pwr (p, n)' which returns the integer value of 'p' raised to the integer power 'n'.
***a)* using recursion** **(6 marks)**
***b)* using iteration** **(6 marks)**

A popular question. Too many candidates used functions without any parameters and had prompts for 'o' and 'n'. No type was generally associated with the function itself. Those who rely exclusively on memory for this kind of question should check it over that some essential stage (like the recursion call) has not been forgotten. A few wrote essays on recursion and/or iteration, which were definitely not wanted.

Answer Pointers

p to the power n or (p ↑ n) by

(a) recursion

```
FUNCTION  pwr ( p , n : INTEGER) : INTEGER;

BEGIN
   IF  n = 0 THEN
      pwr := 1                    {termination : must be present}
   ELSE
      pwr := p * pwr (p , n-1)          {the essential recursive call    }

END;
```

(b) by iteration

```
FUNCTION  pwr (p , n : INTEGER ) : INTEGER;
VAR  ct, temp : INTEGER;

BEGIN
   IF n = 0 THEN
        pwr := 1
   ELSE
        BEGIN
           temp := 1;
           FOR ct := 1  TO  n  DO temp := temp * p;
           pwr := temp
        END
END
```

QUESTION SEVEN

7. Develop an algorithm for a function 'gentemp' which converts temperatures between the two commonly used scales, Celsius and Fahrenheit. The function has two parameters: 'namescale', being 'C' when 'Celsius' is the input temperature, or 'F' when a 'Fahrenheit' temperature is input. Any other entry is an error condition. The second parameter is the temperature (real number) which is to be converted. Use the relationship below to inter-convert the temperatures. (12 marks)

$$\frac{F - 32}{C} = \frac{9}{5}$$

Many did not write a function here, but instead had a simple interactive program that asked for the code letter and temperature value to be input interactively. Again candidates were weak on parameters.
Others lost marks for using the wrong formula in the code, or the 'C' and 'F' formulae interchanged. Both of these were penalised. Simple development even for straightforward problems is desirable. The skill is in knowing how to do the right amount of it, not in leaving it out to save time.

Answer Pointers

The given formula $\frac{F - 32}{C} = \frac{9}{5}$ has to be transformed into versions explicit for calculating 'F' and 'C' separately:

Thus 5 (F - 32) = 9 * C C = (F - 32) * 5 / 9 (expression 1)

and F - 32 = 9 * C / 5 F = 9 * C/ 5 + 32 (expression 2)

Algorithm:

The function ‘gentemp’ is supplied the value to be converted (real, not integer) and a single character which is ‘C’ if this value is on the Celsius scale, ‘F’ if it is on the Fahrenheit scale. Thus

```
IF scale = ‘C’ THEN
    calculate F-value according to expression (2)
ELSE
    IF scale = ‘F’ THEN
        calculate C-value according to expression (1)
    ELSE
        error message
```

Even an algorithm this simple is worth writing as it makes clear which formula is to be used where in the code.

```
FUNCTION  gentemp (scale : CHAR; t : REAL) : REAL;

BEGIN
  IF scale = ‘C’ THEN
      gentemp := 9 * t / 5   +  32                    {returns Fahrenheit
temperature}
  ELSE
      BEGIN
        IF scale = ‘F’ THEN
            gentemp := (t - 32) * 5 / 9               {returns Celsius
temperature}
        ELSE
            WRITELN (‘incorrect scale indicator’)
      END
END;
```

QUESTION EIGHT

8. ***a)*** **For each of the following file organisations, outline two business applications for which the file organisation is suitable.**

i) **Sequential file organisation** **(3 marks)**
ii) **Random file organisation** **(3 marks)**

b) **Sketch the index structure for each of these file organisations and describe how insertions, deletions and updates are implemented by this indexing structure.** **(6 marks)**

Answer Pointers

a) A business purpose for sequential file organisation is typically batch-oriented processing such as payroll, transaction recording and invoicing/billing.

A business purpose for random file organisation is typically real-time processing such as telephone enquiry handling and on-line lookup/bookings

b) A simple sketch of the index structure was all that was required.

QUESTION NINE

9. *a)* Web pages often offer a 'frames' and a 'no frames' option. Explain what these terms mean, and why they are important. (4 marks)

***b)* Sketch the graphical user interface (GUI) for a 'frames' web presentation, and give reasons for <u>two</u> of the GUI features you have included. (8 marks)**

Answer Pointers

a) Frames and No-Frames are GUI layouts that are supposed to aid navigation within a website.

The basic unit of web access is 'the page' and *frames*, multiple windows or pages on a screen, seem to contradict the web's initial simplicity.

b) Two structures and reasons similar to the following are acceptable

Print: a mechanism to produce 'printer-friendly' material well-separated from other-frame-based material on-screen.
Home: always have a mechanism to return to the primary URL; avoids being 'trapped' without an effective 'back' button.
Bookmarking: a mechanism to capture URLs
NO-FRAMES switch for older browsers

QUESTION TEN

10. You have been asked *to test* an interactive web site that has been already implemented. The web site can capture clients' names and email addresses (when they complete answer-back forms) as well as offering to known clients facilities such as downloads of specialist advice relevant to your employer's main business area.

Briefly describe FOUR issues relevant to this kind of web presentation where the testing must be planned and completed satisfactorily before going 'live'. (12 marks)

Answer Pointers

An interactive website should have the following functions properly operating.

- Copyright has been obtained on all materials in the website
- Hot links all working
- Domains and trade marks are used legally
- Companies Act compliance for identification of owner/operator
- Suitable for access on a variety of browsers
- 'Clean' use of meta-tags, no misrepresentation as to contents of website
- User consent for acquisition of personal data (capturing clients' emails and holding a database of clients' details)
- Specialist advice checked with lawyers for defamation, liability and indemnity.

QUESTION ELEVEN

11. Describe, with diagrams, the operation of a look-up table and identify an application that is particularly suited to this form of search and retrieval. Give reasons for your choice. (12 marks)

Answer Pointers

The example of use must be a relatively stable, non-changing dataset e.g. a stable phone-list, or rental rates for a car hire firm. What is not appropriate is some dynamic population of data, for then the prepared key and index mechanism with collision avoidance won't work at all well. For example, a set of customers, a set of orders, or a set of bookings.

The key K is some identifier of the desired data (DD). Suppose L is the table(1:n;1:2).

Then the pointer L(K, 2) is what points to the DD. Generation of K is either directly from the DD or some previously-coded K derived by transformation from DD.

A null pointer, L(K, 2) = NULL, means no data, i.e. DD does not exist in this scheme.

A non-match, K <> L(K, 1), means at least two DD elements have the same transformation; i.e. the same K.

Strategies to combat this are broadly to live with it or avoid it. *Live with it* means implementing some form of scan to find the true presence/absence of K. Common techniques are sequential scanning or some hybridised, randomised skip. It is important to prepare the table for the set of DDs to be looked-up, so that the collisions are anticipated and the avoiding/re-search mechanism works with acceptable performance.

Avoid it means find another way that has far fewer collisions when DD is transformed into K.

QUESTION TWELVE

12. Write brief notes on each of the following:

a) **independent compilation** **(4 marks)**
b) **formal and semi-formal specification of algorithms** **(4 marks)**
c) **search engines** **(4 marks)**

Answer Pointers

a) independent compilation should contain compilation by parts, pieces later to be linked together, productivity from multiple module production, and sequence dependencies of compilation for interface checking should be mentioned.

b) formal and semi-formal specification of algorithms: the formal approach is more aimed at an algebraic framework of verifiable reasoning about the behaviour of the modelled algorithm. Tools and notations of note are Z, B-Tool, Vienna Development Method (VDM).

The semi-formal specification is associated with diagrammatic models of behaviour and functionality, such as OO diagrams, ERDs, state-transition diagrams and data flow diagrams. The aim of these specifications is to capture the desired functionality for programmers to implement. Retrospective walkthroughs and compliance tests are conducted as part of the verification activities.

c) Meta-search is the name given to a search engine that uses several other search engines to make its searches for it. In addition, a meta-search engine often identifies which other search engines it is using and keeps a database record of searches performed and results found. Some meta-search engines update the set of real engines used as part of normal processing.