## THE BCS PROFESSIONAL EXAMINATION

## Professional Graduate Diploma

## March 2017

## EXAMINERS' REPORT

# SOFTWARE DEVELOPMENT

### General comments on candidates' performance

*The standard for this examination was reasonable, with many candidates gaining high marks.*

*However, although there were some excellent papers, a large number of candidates lost marks needlessly by providing answers that contained minimal content or were unrelated to the question. Candidates are therefore advised to spend more time reading and understanding the requirement of each question before writing their answers. In addition, many candidates answered more than the required number of questions. Future candidates should note that no credit is given for answering additional questions.*

*Please note that the answer pointers contained in this report are examples only. Full marks were given for alternative valid answers.*

## SECTION A

(Candidates were required to answer **TWO** out of the four questions set)

### A1

Imagine data obtained by reading a thermometer every hour for a week. The problem considered here is to find where the temperature peaks and to measure the gaps between the peaks.

a) Write a function (called findDirection) to go element by element through an array (called TEMPERATURE) and place in a new array (called DIRECTION) a +1 every time that a value in the first array is greater than the previous value, a 0 if it is the same and -1 otherwise.

(**8 marks**)

```
TEMPERATURE   21  22  22  23  22  21  21  22  23  24  25  23  21  22  21  22
DIRECTION         +1   0  +1  -1  -1   0  +1  +1  +1  +1  -1  -1  +1  -1  +1
```

b) Write a function (called change0) to go through the array called DIRECTION) changing every 0 to a copy of the preceding entry.

(**8 marks**)

```
index        0   1   2   3   4   5   6   7   8   9  10  11  12  13  14 ...
DIRECTION       +1   ~~0~~  +1  -1  -1   ~~0~~  +1  +1  +1  +1  -1  -1  +1  -1  +1
                    +1                  -1
```

c) Write a function (called findPeaks) to record in a new array (called PEAKS) the indexes in DIRECTION where the peaks are [where the temperature changes from going up (+1) to down (-1)].

(**8 marks**)

```
index        0   1   2   3   4   5   6   7   8   9  10  11  12  13  14 ...
DIRECTION       +1  +1  +1  -1  -1  -1  +1  +1  +1  +1  -1  -1  +1  -1  +1
                          ^                           ^           ^
PEAKS 3, 10, 13
```

d) The data is provided in a 168(=24hrs*7days) element array called TEMPERATURE. Write a program, using the functions of parts (a), (b) & (c) to report the lengths (in hours) of the gaps between peaks. (**6 marks**)

**Answer Pointers**

```
int TEMPERATURE[168]={values predefined};
int DIRECTION[168];
int max=168;
void findDirection(){
      int i;
      for(i=1;i<max;i++){ //start at 1 so subscript i-1 has meaning
            int t0=TEMPERATURE[i-1];
            int t1=TEMPERATURE[i];
            if(t1>t0) DIRECTION[i]=1;
            else if(t1==t0) DIRECTION[i]=0;
            else DIRECTION[i]=-1;
      }
}
void change0(){
      int i;
      for(i=2;i<max;i++){ //start at 2 so subscript i-1 has meaning
            if(DIRECTION[i]==0)
                  DIRECTION[i]=DIRECTION[i-1];
      }
}
int PEAKS[168];
void findPeaks(){
      int p=0;
      int i;
      for(i=1;i<max-1;i++){ //start at 1 so subscript i+1 has meaning
            if(DIRECTION[i]==1 && DIRECTION[i+1]==-1){
                  PEAKS[p++]=i;
            }
      }
}
int main(){
      int i;
      findDirection();
      change0();
      findPeaks();
      for(i=0;i<max;i++){
            printf("%d ",PEAKS[i]);
      }
}
```

**Examiners' Guidance Notes**

*Although this wasn't a popular question in section A, many of the candidates that attempted the question produced a correct or near correct solution for part (a) where functions were created to find the direction of temperature change of the 168 array elements stored.*

*In part (b) functions were used assign 0 values in the direction array to the preceding entry and in part (c) functions were used to create an array termed "Peaks", where temperature change between (+1) and (-1) was recorded.*

*Many candidates lost marks as they did not attempt part (d) or failed to include the correct code to report the length in hours between peaks.*

**A2**

There are 6 candidates in an election and the voting paper has been designed as follows:

```
Mrs Adams [ ] Mr Bean [ ] Mr Crab [ ] Mrs Dean [ ] Mrs East [ ] Mr Fish [ ]
```

The voters have been told to put a number 1 next to their first choice, a number 2 next to their 2nd choice, and so on down to a 6 for their 6th choice. The scoring system is that a candidate is awarded 6 points every time they are chosen in 1st place, 5 points for 2nd, down to 1 point for 6th. You are to write a program to read the choices data, apply the scoring system and report the winning candidate (the candidate with the most points).

The data is available to be read from a file called CHOICES containing one line for each voter. Each line contains just the numbers (always six numbers) from one voting form so

5 4 1 6 3 2

would mean that a voter put Mr Crab in 1st place and Mrs Dean last (in 6th place). The data file contains the choices from 100 voters. You can assume that an array called CANDIDATES exists which contains the candidates' names in alphabetical order.

(**30 marks**)

*[Note: For maximum marks your answer should use functions and should produce a warning message if there is a tie for 1st place.]*

**Answer Pointers**

Use of functions (e.g. readAndTotal, MaxAndTie) 10 marks (5 marks each)
Discovering and reporting a tie: 5 marks
Note: This answer does NOT use functions

```
for(c=1;c<=6;c++){ //c for candidate
      POINTS[c]=0;
}
for(v=1;v<=100;v++){ //v for voter
      for(c=1;c<=6;c++){ //c for candidate
            scanf("%d",&place);
            POINTS[c]+=7-place;
                        //place=1 is awarded 6 points, 2nd=5, ... 6th=1
      }
}
winner=1; //start by assuming that the winner is in position 1
tie=0;
for(c=2;c<=6;c++){ //c for candidate, look through rest of candidates
      if(POINTS[c]==POINTS[winner])
            tie=c;
      if(POINTS[c] > POINTS[winner]){ //we have a new winner
            winner=c;
            tie=0;
      }
}
printf("The winner is %s.",CANDIDATES[winner]);
if(tie){
      printf("There is at least one other candidate with the same total %s.",
                  CANDIDATES[winner]);
}
```

(30)

**Examiners' Guidance Notes**

*This was the least popular questions in Section A. In a few cases candidates produced a structured solution to gain a pass mark.*

*However, the evidence suggest that few marks were gained by the majority of candidates as the quality of answers was generally poor; for example, many candidates entered a list of assignments for each candidate with votes based on the form given in the question.*

**A3**

a) Give the final values of the variables a, b, c, d after the following code has executed:

```
i=2; a=++i; b=i;
i=4; c=i++; d=i;
```

(**4 marks**)

b) Based on your answer to part a), or otherwise, state the key difference between ++i and i++.

(**4 marks**)

c) Given the initial values in the arrays S1 and S2 as follows:

```
index 0   1   2   3   4   5
S1    11  12  15  16  19  22
S2    13  14  17  18  20  21
```

trace the execution of the function call ***f***(3), where the function ***f*** is defined as follows:

```
void f(int g){
      int i1=0, i2=0, i3=0;
      while(i1<g && i2<g){
            if(S1[i1]<S2[i2]){
                  S3[i3++]=S1[i1++];
            }else{
                  S3[i3++]=S2[i2++];
            }
      }
}
```

(**18 marks**)

d) Describe in your own words the kind of data that function ***f*** has been given and describe the result that it produces.

(**4 marks**)

**Answer Pointers**

a) a=3, b=3, c=4, d=5 (4)

b) both ++i and i++ have the effect of incrementing the value of i by 1, but when ++i occurs inside a larger expression the value delivered is the incremented value, whereas the delivered value of i++ is the original value of i (before incrementing). (4)

c) trace:

```
g=3
                    index 0   1   2   3   4
         i1  i2  i3    S3
         0   0   0
11<13 T  1   0   1        11
12<13 T  2   0   2        11  12
15<13 F  2   1   3        11  12  13
```

```
15<14 F  2   2   4         11  12  13  14
15<17 T  3   2   5         11  12  13  14  15
```

while loop now stops because i1 is too big (18)

d) f has to be given arrays in ascending order and its result is the merge the two arrays into a single array also in ascending order. The parameter g measures the length of the input arrays to be processed. (4)

**Examiners' Guidance Notes**

*Although this was a popular question attempted by 84% of the candidates, less than 20% achieved a pass mark.*

*Few candidates produced a correct or near correct solution, the main problem areas were:*

- *Parts (a and b) – the evidence suggests that mistakes were made as candidates were confused by the differences between ++i and i++*
- *Parts (c and d) – there were frequent errors in tracing the function f(3), consequently the candidates were unaware that the two arrays were merged into one array in ascending order.*

**A4**

Based on the following program extract, answer the questions below:

```
#include <stdio.h>
#include <math.h>
void f(char c, float y){
      int r; r=0; y=0.0;
      if(c>='0'&&c<='9'){r+=10; y=sqrt(r);}
      printf("%d %f", r, y);
}
```

From the extract

a) List all the identifiers (**5 marks**)
b) List all the operators. (**5 marks**)
c) List all the constants. (**5 marks**)
d) List the types of each of the constants in c). (**5 marks**)
e) Copy out an example of each of the following:
   i) a declaration,
   ii) a boolean expression,
   iii) an assignment statement,
   iv) a conditional statement. (**10 marks**)

**Answer Pointers**

a) f, c, y, r (5)

b) =, >=, <=, &&, += (5)

c) 0, 0.0, '0', '9', 10, "%d %f" (5)

d) int, float, char, char, int, string (5)

e) declaration: int r
boolean expression: '0'<=c
assignment statement: r=0;

conditional statement: if('0'<=c&&c<='9')r+=1; (10)

### Examiners' Guidance Notes

*This was an extremely popular question that was attempted by 94% of the candidates; it was generally well answered with approximately 75% of the candidates achieving a pass mark.*

*Most of the candidates were able to gain maximum or near maximum marks for parts (a, b) and generally all showed a reasonable understanding of part (e).*

*Marks were mainly lost in part (c, d) where many candidates failed to identify all of the constants and the types of constant used in the code.*

## SECTION B

**Answer 5 questions (out of 8). Each question carries 12 marks.**

**B5:** Values for the hyperbolic sine function are obtained from the power series

**Sinh(x) = x + $x^3$ / fac(3) + $x^5$ / fac(5) + $x^7$ / fac(7) + …**

Where: fac(n) = factorial n = 1*2*3*4*…n

Note: - Use pseudocode or actual program code of your choice to answer this question.

a) Write code for fac(n); any method may be used.

**(4 marks)**

b) Incorporate your function into another function HSine(x) which calculates Sinh(x) using the power series given above. The calculation should be terminated when the difference between successive terms is less than 0.00005.

**(8 marks)**

### Answer Pointers

a) Accept iteration or recursion method for factorial code

```
FUNCTION Factorial (N: INTEGER): INTEGER
{Using recursion method}
        IF (N = 0)
                Factorial ← 1;
        ELSE
                Factorial ← N * Factorial (N-1);
        END
END FUNCTION
```

b)

```
FUNCTION HSine(x)

        Accuracy ← 0.00005;   // Initialisation
        Sum ← x;
        N ← 1;
        Multiplier ← x;
        Past Term ← x;
        Difference ←1;

        WHILE (Difference > Accuracy) DO
                Multiplier ← Multiplier * x * x;
                N ← N + 2;
                Term ← Multiplier / Factorial(N);
                Sum ← Sum + Term;
                Difference ← Abs(Past Term - Term);
                Past Term ← Term;
        END WHILE

        HSine ← Sum;

END FUNCTION
```

**Examiners' Guidance Notes**

*This was the least popular question attempted by only 11% of the candidates, with around a third achieving a satisfactory pass mark.*

*Few candidates produced a correct or near correct solution, the main problem areas were:*

- *Generally candidates produced suitable algorithms, particularly for part (a) but often produced excessive code with far too many logical errors. Part (b) – the evidence suggests that candidates did not have the knowledge to express a function (e.g. fact) within the main body of another function (sinh).*

**B6:** The following code extracts are written in C;

a) Write code using a while loop which is equivalent to the following *for* loop.

```
for(i=0;i<=99;i++)
   v[i]=0;
```

–

b) Rewrite the following conditional code without using the logical operators &&(and), ||(or), !(not)

```
if( p && !q )
   x = 0;
else
   x = 1;
```

c) Write code to find the maximum value in array *v* with the answer in variable *max*.

```
int v[100];
int max;
```

–

**(3 x 4 marks)**

**Answer pointers**

a)

```
i=0;
while(i<=99){
        v[i]=0;
        i++;
}
```

b)

```
if(p)
        if(q) x=1;
        else x=0;               // p && !q --> 0
else
        x=1;
```

c)

```
int i, max, v[100];
...
max=v[0];
for(i=1;i<100;i++)
        if(v[i]>max)
        max=v[i];
```

**Examiners' Guidance Notes**

*This was a popular question attempted by 76% of the candidates with more than 83% achieved a pass mark. The best performance from Section B.*

*Many candidates produced a correct or near correct solution for. The evidence suggests that in part b) many candidates did not understand the Boolean logic in the supplied code. The Logical AND, NOT operators were often recast as simple IF THEN ELSE statements.*

**B7:** Describe the following file types:

a) Sequential access file
b) Direct access file
c) Indexed sequential file

**(3 x 4 marks)**

**Answer pointers**

a) A sequential access file is one where the record is stored in an ordered or pre-determined sequence. The only way to locate a particular record is to start at the beginning of the file and read all the preceding records up to the location of required record. Sequential access files are often stored on serial access devices such as magnetic tape and can be used efficiently for backing up data and batch processing.

b) A direct access file is also known as a random access file and data is organised using an index. This enables quick access to individual records within the file as the index points to a specific location within the file and the record data is read directly from that location. In direct access files any record can be accessed in approximately the same time.

c) Records in indexed sequential files are stored in the order that they are written to the disk; each record is composed of fixed length fields. Indexes are used to speed up access to read and write to the file and in the case of an employee data base the key filed or index could be an employee number or name.
A separate table of indexes is used which contains pointers to specific locations in the original sequential file; this allows individual records to be retrieved without having to search the entire file; which improves the speed of access when compared to a sequential file.

**Examiners' Guidance Notes**

*This was a popular question attempted by 80% of the candidates, 57% achieved a pass mark.*

*A number of candidates produced a correct or near correct solution, the main problem areas were:*

- *Overall the evidence suggests that candidates had limited knowledge and wrote simplistic answers with little expansion on the concepts involved. In particular candidates failed to get full marks due to a lack of examples, for example how these file types are applied in practice.*
- *Part c) Although very few mistakes were made overall many candidates could not explain the characteristics of indexed sequential files.*
- *The evidence suggests that many candidates understood concepts but did not add any detailed explanation.*

**B8:** a) Explain the difference between a syntax error and a run-time error? **(3 marks)**

b) Write down an example for each of the three errors listed below. In each case examples should be a one line extract from a program with a precise description of the error.

i) A syntax error in an expression **(3 marks)**

ii) A syntax error in a statement **(3 marks)**

iii) A run-time error in an expression **(3 marks)**

**Answer Pointers**

a) **Syntax errors** occur at compile time, the source code needs to conform to the syntax of the programming language or the compilation process cannot be completed.
**Run-time errors** occur during the execution of the program; typical run-time errors are running out of memory or an unusual numerical situation has occurred e.g. divide by zero

b) i) In expression **( 2 / * 3 )** – there is a syntax error as only one (binary) operator required between operands

ii) In statement **if( x>1 {y=1;}** there is a syntax error as the closing parenthesis is missing after x>1

iii) In expression **a / (b - b)** there is a run-time error for division by zero

**Examiners' Guidance Notes**

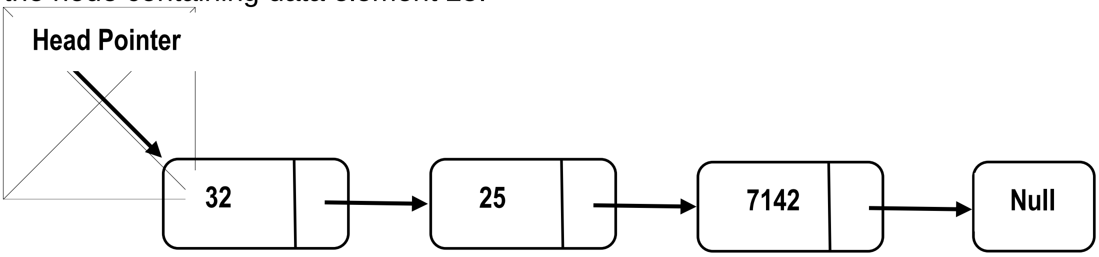
*This was a popular question attempted by 87% of the candidates, with 60% achieving a pass mark.*

*Some candidates produced a correct or near correct solution, with part b) generally well answered. However the main problem areas were:*

- *Part (b) – Many candidates could not differentiate between syntax errors in expressions with syntax errors in statements so incorrect code examples were common.*
- *Some candidates wrote code or pseudo code without any explanation which was marked down.*
- *Longwinded segments of code were not required and this resulted in marks being lost.*

**B9.** A linked list has been set up with one data item and one pointer only in each element of the list as shown in the diagram below.

a) Draw a diagram of this linked list with a new node inserted containing data element 621 after the node containing data element 25. **(2 marks)**



b) Write pseudocode for a function **InsertAfter**, with appropriate parameters, to search for a data item held in the key node and to insert the new node with data element equal to 621.

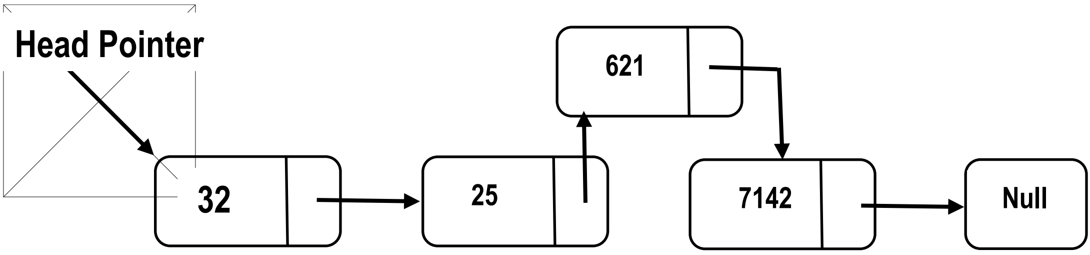
**(6 marks)**

c) Describe the linked list data structure giving an advantage of using this technique.

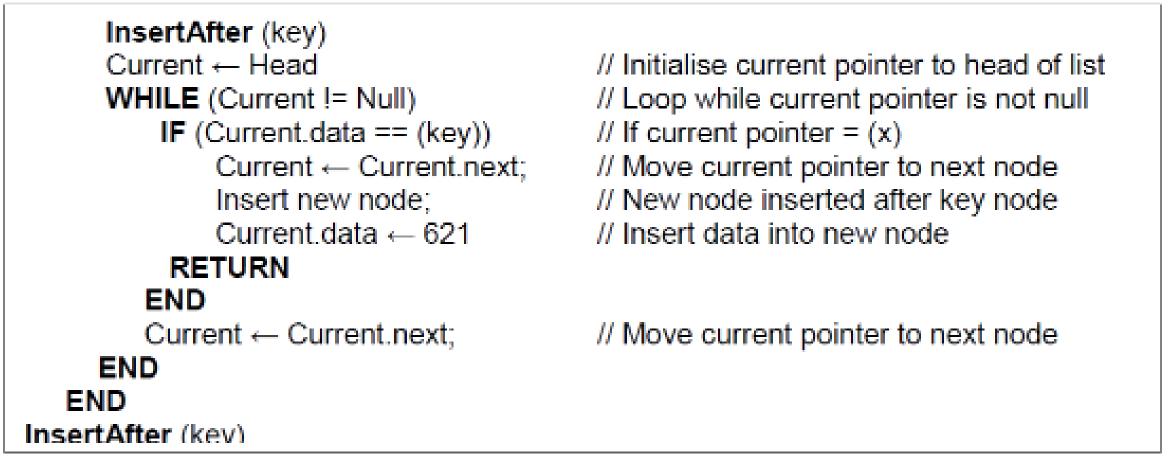
**(4 marks)**

**Answer Pointers**

a) Linked list diagram



b) **InsertAfter** pseudocode solution.



```
            InsertAfter (key)
            Current ← Head                  // Initialise current pointer to head of list
            WHILE (Current != Null)         // Loop while current pointer is not null
                IF (Current.data == (key))  // If current pointer = (x)
                    Current ← Current.next; // Move current pointer to next node
                    Insert new node;        // New node inserted after key node
                    Current.data ← 621      // Insert data into new node
                 RETURN
               END
               Current ← Current.next;      // Move current pointer to next node
           END
         END
     InsertAfter (key)
```

c) A linked list is a linear data structure in which a group of nodes contain data stores data the following way:
It creates a "list item" for each data item and holds the actual data item and a pointer to the next "list item". If it is doubly linked, it also contains a pointer to the previous "list item":
Linked-lists are dynamic structures as the number of nodes required is not fixed, so that it can be resized to fit the requirements of an application.
Linked lists have advantages when it comes to inserting at random positions
If your operation involves adding lots of elements dynamically and traversing all elements
Because linked lists are good for inserting records, they can be used as underlying data structures for search trees in which the lowest level will consist of a linked list.

**Examiners' Guidance Notes**

*This question was along with Q5, one of the least popular questions in Section B, attempted only by 10% of the candidates, with only a third of candidates achieving a pass mark.*

*However, there was evidence to suggest that a small number of candidates produced a correct or near correct solution, the main problem areas were:*

- *Part (b) Many candidates failed to produce suitable code, in particular candidates did not consider the dynamic nature of a link list and as result wrote very generalised code rather than the specific code that was required to insert the given data item.*
- *Part (c) – The evidence suggests that there was a lack of knowledge of this data structure.*

**B10.** Write notes to compare and contrast the following pairs of terms

a) relational and flat file databases
b) system software and application software
c) black box and white box testing

**(3 x 4 marks)**

**Answer Pointers**

**a)** In a **relational database** the data is contained in more than one table. Tables are then joined by linking a primary key in one table to a foreign key in another table. In a **flat file database** the data is stored in a single table. Flat file databases are generally in plain-text form, where each line holds only one record. The fields in the record are then separated using delimiters such as commas and tabs.

**b)** **Systems software** is a type of computer program that manages and operates the hardware to provide a platform for the operation of application software. **Applications software** is a type of computer program designed to perform a series of tasks or activities to benefit the end user; examples include word processor and web browser.

**c)** **White box** testing a process of testing a program with full knowledge of the code, so that the testing can guarantee to exercise all pathways through the code. **Black box** testing a process of testing a program without full knowledge of the code, so that the testing checks that results for particular input match the specification.

**Examiners' Guidance Notes**

*This was the most popular question attempted by 97% of the candidates, with 78% achieving a satisfactory pass mark.*

*Many candidates produced correct or near correct solutions, with part b and c) generally well answered by most candidates. Evidence suggests that the main problem areas were :*

- *Part (a) – Some candidates could not explain the concept of a flat file and therefore found it hard to differentiate it.*
- *Candidates should provide illustrative examples of different uses and benefits of each topic and avoid longwinded answers.*

**B11**: Describe the following stages of compilation:

a) Lexical analysis

(4 marks)

b) Syntax analysis

(4 marks)

c) Code generation

(2 marks)

d) Code optimisation

(2 marks)

**Answer pointers**

a) Lexical analysis is the first stage of compilation where the high level input code is broken down into a form suitable to be analysed in the next stage of compilation. White space and comments are removed from the code leaving a series of tokens (strings of characters with an identified meaning). These tokens include: reserved words, constants, variables and operators.

b) The second stage of compilation is syntax analysis which consists of determining whether the strings of input tokens form a valid sentence in the grammar of the programming language used. At this stage a dictionary of the variables used in the program is generated containing the location of these variables in memory.

c) The third stage of compilation is code generation, which is where machine code is created for the target machine. Each high level language instruction may consist of several low level or machine code instructions.

d) Code optimisation is the final stage of compilation where the machine code generated is optimised to make it more efficient. Inefficient code can be optimised by replacing certain instructions or groups of instructions with more effective alternatives.

**Examiners' Guidance Notes**

*This was a fairly unpopular question attempted by 26% of the candidates, with 30% achieving a pass mark.*

*Very few candidates produced correct or near correct solutions, with part a) poorly answered by most candidates. The evidence suggests that the main problems were:*

- *Part (a) – Most candidates could not explain the principles behind lexical analysis and either produced incorrect answers or failed to produce any answer.*
- *Part (b) – Candidates showed a lack of knowledge and understanding of the process of compilation evidenced by a lack of illustrative examples of processing each stage of compilation covered in the question.*

**B12:** Briefly describe the operation of the following methods for software development and the particular advantages of using them.

a) Traditional waterfall method **(6 marks)**

b) Software prototyping **(6 marks)**

**Answer Pointers**

a) In the "Waterfall" approach, the whole process of software development is divided into separate phases. In Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially; the phases include: requirements analysis, design, implementation, testing and maintenance.

The advantages of using the waterfall method are:

- Easy to manage due to the rigidity of the model
- Each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Clearly defined stages and well understood milestones.

b) Software prototyping operates in a similar way to the traditional waterfall method except that the developer repeatedly loops through the Requirements, System Design and Coding until a satisfactory solution is obtained.

The advantages of using software prototyping are:

Software prototyping operates in a similar way to the traditional waterfall method except that the developer repeatedly loops through the Requirements, System Design and Coding until a satisfactory solution is obtained.
The advantages of using software prototyping are:

- Ideal for use where the users are unable to accurately specify their information system requirements
- Improves communication between system developers and client
- Helps to identify confusing functions or even missed functionality
- Reduces risk of failure, as possible risks are identified quickly and precautions can be taken to solve the problem.
- Ideal approach for developing software by non-computing specialists

**Examiners' Guidance Notes**

*This was one of the most popular questions in Section B, attempted by 92% of the candidates, with 78% achieving a pass mark.*

*The evidence suggests that a large number of candidates produced correct or near correct solutions, with both parts equally well answered by most candidates. The main problem areas were:*

- *Part (a , b) – Many candidates concentrated on writing memorised material on these topics but failed to balance this with an explanation of the advantages and disadvantages. This was the main area where marks were lost.*
- *Part (a) Some candidates produced over-long answers with little substantive content.*