# BCS THE CHARTERED INSTITUTE FOR IT

## THE BCS HIGHER EDUCATION QUALIFICATIONS
BCS Level 4 Certificate in IT

## SOFTWARE DEVELOPMENT

Tuesday 27th March 2012 - Afternoon
Time: TWO hours

Section A and Section B each carry 50% of the marks. You are advised to spend about 1 hour on Section A (30 minutes per question) and 1 hour on Section B (12 minutes per question).

**Answer the <u>Section A</u> questions you attempt in <u>Answer Book A</u>**
**Answer the <u>Section B</u> questions you attempt in <u>Answer Book B</u>**

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

Calculators are **NOT** allowed in this examination.

**When you are asked to write code, please state the language you are using.**

## SECTION A
Answer TWO questions out of FOUR in Answer Book A.  Each question carries 30 marks

A1.

a)   Write a section of code to read an array *v* (starting at the first element) and which copies all the entries of value 5 or more from array *v* to consecutive locations in array *w*.

Example
Array *v*

| 3 | 9 | 5 | 1 | 2 | 6 | 4 | 8 |
|---|---|---|---|---|---|---|---|

Array *w*

| 9 | 5 | 6 | 8 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|

**(8 marks)**

b)   Convert the code you have written in part a) into a function (or procedure) called *closeUp* which has a parameter called *min* such that no elements of *v* below min are copied to *w*.  The result of the function is the number of elements copied to *w*. (In the case of a procedure, return the number of elements copied to *w* as a second parameter.)

**(6 marks)**

c)   Write a function to find the average of the values in array w from the index *low* to the index *high*

**(8 marks)**

d)   Write a complete program using your answers to b) and c) which reads 100 values from a file into array *v*, calls *closeUp* with parameter 10 and then prints out the average of the non-zero values from array w

**(8 marks)**

A2.  A University department requires a report generator for the progression of students from one year to the next. Write a program to produce this report in line with the following specification.

Students take 12 courses each. The actual data is available in a text file called *results*. Each line in the file corresponds to one student and lists their *student_id* (integer) and their 12 marks (each as a percentage, rounded to an integer). There are 50 students in the class.

The students need to achieve a minimum performance and to gain 100 credits to progress. Minimum performance means scoring 20% or more in every module. Ten credits are awarded for each course for which the student gains 40% or more.

In order to make your program most useful, you should arrange that the key figures in these rules (in this example, 50, 100, 20%, and 40%) are easily changeable in your program so that it can be used with other students for whom the rules are different.

In the resulting report each line should report one student. There should be 6 columns as shown below.

| Student_ id | overall average | total credits | total credit>=100 Yes/No | Minimum performance Yes/No | Progress to next year Yes/No |
|---|---|---|---|---|---|
| 101265 | 55.5 | 100 | Yes | No | No |
| 231498 | | | | | |
| 333444 | | | | | |

**(30 marks)**

A3.  Consider the program segment below and then answer the questions that follow.

| Version in C | Version in Pascal |
|---|---|
| ```
void f(int a){
   int b, c; char d;

   b = 0; c = a - 1;
   while( b < c ){
      d = v[b];
      v[b] = v[c];
      v[c] = d;
      b++;
      c--;
   }
}
``` | ```
procedure f( a : integer );
var   b, c : integer; d : char;
begin
   b := 0; c := a - 1;
   while b < c do begin
      d := v[b];
      v[b] := v[c];
      v[c] := d;
      b := b + 1;
      c := c - 1
   end
end;
``` |

a)  Trace the execution of the call f(8) when the array *v* is initialised as follows. The first element of the array *v* is at index 0.

Array *v*

| 'Q' | 'W' | 'E' | 'R' | 'T' | 'Y' | 'U' | 'I' |
|---|---|---|---|---|---|---|---|

Show the final state of the array v.

**(16 marks)**

b)  State in your own words the overall effect of the function f

**(4 marks)**

c)  Rewrite f making any changes that you think would make the code better for a human reader

**(10 marks)**

A4. Consider the following program and answer the questions that follow.

| Version in C | Version in Pascal |
|---|---|
| ```
char a; float b;
char c(int d, int e){
   float f;

   f = d / e;
   if( f < 1 ){
      b = b + f;
      return( 'y' );
   }else
      return( 'n' );
}
int main(){
   b = 0;
   a = 'y';
   printf( "%c", g );
   printf( "%c", c( 2, 3 ) );
}
``` | ```
program p;
var a : char; b : real;
function c( d : integer; e : integer ):char;
var f : real;
begin
   f := d / e;
   if f < 1 then begin
      b := b + f;
      c := 'y';
   end else
      c := 'n';
end;
begin
   b := 0;
   a := 'y';
   write( a );
   write( c( 2, 3 ) )
end.
``` |

In the program above find and copy out ALL the (different)
a)      integer constants
b)      string or character constants
c)      local variables
d)      global variables
e)      formal parameters
f)      relational operators
g)      type identifiers
h)      actual parameters
i)      arithmetic operators

**(2 marks x 9)**

*Notes: In parts (a-i) only copy out what is asked for - do not include any of the surrounding context.  If an item occurs more than once in the program then it is only necessary to write it down once.*

From the program above, find and copy out ONE example of
j)      an assignment statement
k)      a compound statement
l)      a conditional statement
m)      a function call

**(3 marks x 4)**

B5.    a)    In the programming language of your choice, state how the operations of
            integer division (division with integer result) and modulo (remainder
            after division) are carried out?  State the language used.

**(2 marks)**

       b)    Write a subroutine called *binary* (in the language used in part a) to take an
            integer parameter and output its value in binary.

            For example the call *binary*(29) outputs 11101

**(10 marks)**

B6.    By writing two functions called factorial and factorial_r show how the calculation
       known as factorial may be computed by a non-recursive method and a recursive
       method.

**(12 marks)**

       [ Definition: factorial(n) = n * (n-1) * ... * 2 * 1 ]

B7.    Carry out the following tasks in **either** C **or** Pascal

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

| Version in C | Version in Pascal |
|---|---|
| ```for(i=0;i<=99;i++)```<br>```    v[i]=0;``` | ```for i:=0 step 1 to 99 do```<br>```    v[i]:=0;``` |

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

| Version in C | Version in Pascal |
|---|---|
| ```if( p && !q )```<br>```    x = 0;```<br>```else```<br>```    x = 1;``` | ```if p and not q then```<br>```    x := 0```<br>```else```<br>```    x := 1``` |

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

| Version in C | Version in Pascal |
|---|---|
| ```int v[100];```<br>```int max;``` | ```var v:array[0..99] of integer,```<br>```    max:integer;``` |

**(3 x 4 marks)**

B8.    Answer the questions by referring to **either** the C **or** the Pascal version of the code
       below:

| Version in C | Version in Pascal |
|---|---|
| ```int f(int j,int k){int```<br>```i,m;m=0;for(i=0;i<=```<br>```7;i++)if(j<=v[i]&&v[i```<br>```]<=k)m++;return m;}``` | ```function f(j,k:integer):```<br>```integer;var i,m:integer;```<br>```begin m:=0;for i from 0 to```<br>```7 do if j<=v[i] and v[i```<br>```]<=k then m:=m+1;f:=m end``` |

       a)    Comment on the presentation/layout of this code?

**(2 marks)**

b)    Rewrite the code to make it look more conventional.

**(4 marks)**

c)    Could the original 'bad' code run successfully without the change you made in b)? Why?

**(2 marks)**

d)    Explain in your own words what the code is trying to achieve.

**(2 marks)**

e)    Apart from what you did in b), suggest one more improvement to the code.

**(2 marks)**

B9.    a)    What do you understand by the term 'debugging'?

**(4 marks)**

b)    In a simple programming environment where the programmer has only the standard output facilities of the programming language to use, how is debugging approached?

**(4 marks)**

c)    What extra facilities to assist in debugging might be provided in a more extensive development environment?

**(4 marks)**

B10.    Supply the words to complete the following sentences:

a)    To translate and run a program a _____ or an _____ is needed.
b)    The first 2 phases of the Software Development Life Cycle are _____ & _____.
c)    The acronym WIMP stands for _____, _____, _____, _____.
d)    The list of common programming language paradigms includes functional languages, _____ & _____.
e)    Depending on the hardware device involved, a computer program may have _____ access or _____ access to a file.
f)    A computer program is translated from _____ language to _____ language.
g)    The two kinds of buttons that tend to occur in groups on web forms are called _____ & _____.
h)    The acronyms LIFO and FIFO stand for _____ & _____.
i)    The basic control structures found in computer programs are sequencing, _____ & _____.
j)    The basic (or primitive) data types in a programming language include integer, _____ & _____.
k)    Apart from basic data types there are constructed types which include _____ & _____.
l)    Two popular methods of testing programs are _____ & _____.

**(1 mark x12)**

B11.    a)    Name two algorithms that can be used for sorting an array.

**(2 marks)**

b)    Choose ONE of the algorithms named in a) and describe how it works.

**(10 marks)**

*(Note: code is not required)*

B12.  a)  What is the difference in programming terms between an array and a record?

**(3 marks)**

  b)  Give typical examples of
    i) data that would be handled as an array
    ii) data that would be handled as a record
    iii) data that would be handled as a combination of array and record

**(3 x 3 marks)**