

BCS THE CHARTERED INSTITUTE FOR IT
BCS HIGHER EDUCATION QUALIFICATIONS
BCS Level 4 Certificate in IT

SOFTWARE DEVELOPMENT

Wednesday 18th September 2019 - Morning

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 any Section A questions you attempt in Answer Book A
Answer any Section B questions you attempt in Answer Book B

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

Calculators are NOT allowed in this examination.

[Turn Over]

SECTION A

Answer 2 questions (out of 4) in Answer Book A. Each question carries 30 marks.

A1. An array **arr** contains 100 integers.

- a) Write a function called *reverse* that will take **arr** and a second 100 element integer array **revarr** as parameters and copies all the numbers in **arr** into **revarr** in reverse order.

(10 marks)

- b) Write a second function that takes **arr** as a parameter along with two other arrays named **odd** and **even**. The function should copy all the odd numbers from **arr** to the array **odd** and copy all the even numbers from **arr** to the array **even**.

(10 marks)

- c) Write a function that takes **arr** and a variable **s** containing a single integer as parameters. Your function should return a variable that contains the number of times that **s** occurs in **arr**.

(10 marks)

A2.

- a) Write a program to enable users to play a guessing game where the user has to guess a secret number generated by the program. After every guess the program tells the user whether their number was larger than the secret number or smaller than the secret number. The game is over when a user inputs a number equal to the secret number. At the end of the game, the number of tries needed to obtain the correct number should be printed. Your program should make use of an existing function called *randint* that will generate a suitable secret number.

(10 marks)

- b) Write a program that uses the function *randint*, referred to in A2 a), to generate each of the four digits in a 4-digit sequence and then invites the user to enter their own 4-digit sequence as a guess. Your program should tell the user how many digits they guessed correctly in the correct place and how many digits have been guessed correctly but in the wrong place. The user is then invited to change their guess based on the information they have been given until they get the correct answer.

(20 marks)

[Turn Over]

A3. Consider the following code:

```
s = 0;
c = 0;
input age;
while (age <> 0){
    s = s + age;
    c = c+1;
    input age;
}
a = s / c;
print a;
```

- a) Suggest comments that could be added to this code to explain what the program is doing and suggest better names for variables s, c and a. Justify your suggestions. **(10 marks)**
- b) Explain, with examples, how appropriate use of variable names can help the programmer write self-documenting code. **(5 marks)**
- c) Rewrite the code above using a repeat-until-loop instead of a while-loop. **(5 marks)**
- d) Under what circumstances should you use a for-loop instead of a while-loop? **(5 marks)**
- e) Do programmers need to use repeat-until-loops or should they always use while-loops instead? Explain your reasoning. **(5 marks)**

A4. Write code which implements a simple calculator that prompts the user for 2 numbers, and then asks the user which operation they wish it to perform. The calculator must be able to do the following operations:

- i) Addition;
- ii) Subtraction;
- iii) Multiplication;
- iv) Division;
- v) Exponentiation (raising one number to the power of another).

You can assume that users of the code will only enter numbers that are integers greater than zero.

After printing the result of an operation, the calculator must ask the user if they want it to perform another calculation and do so if they do. The program should terminate when the user says they do not want to perform any more calculations.

(30 marks)

[Turn Over]

SECTION B

Answer 5 questions (out of 8) in Answer Book B. Each question carries 12 marks.

- B5.** The Fibonacci series of numbers begins with the integers 0 and 1 and each succeeding number in the series is calculated by adding together the two preceding terms. For example:

$0+1 = 1$. $1+1 = 2$; $1+2 = 3$; $2+3 = 5$; $3+5 = 8$ and so on, resulting in the sequence 0, 1, 1, 2, 3, 5, 8,

To check whether a given integer (n) is a Fibonacci number the following algorithm is used:

Algorithm

If n is a Fibonacci number then the result of *EITHER* $(5n^2 + 4)$ *OR* $(5n^2 - 4)$ must be a perfect square (a perfect square is a number that can be expressed as the product of two equal integers).

[For example, if $n=4$ the above formula returns 76 and 84 respectively, neither of which are perfect squares. Therefore 4 is NOT a Fibonacci number].

- a) Show by working through the above **algorithm** that the number 3 *is* a Fibonacci number. **(2 marks)**
- b) Using pseudocode or actual code:-
- i) Write a function called *ISperfectsquare(n)* that returns TRUE if an integer n is a perfect square otherwise it returns FALSE. **(5 marks)**
- ii) Write another function called *ISfibonacci(x)* that calls *ISperfectsquare(n)* and returns TRUE if x is a Fibonacci number, otherwise FALSE. **(5 marks)**

B6.

- a) Draw a flow chart for a program that inputs three integers into three variables and outputs the value of the largest of the three integers. **(4 marks)**
- b) Translate your flow chart into pseudocode or actual code. **(4 marks)**
- c) Define **Black Box** Testing and explain using examples how you would test your code using **Black Box** Testing. **(4 marks)**

[Turn Over]

B7. Suppose you are one of a team of 20 programmers. The team works on a large project with each programmer given an individual programming task to complete.

a) List FOUR pieces of documentation that each programmer should produce on completion of their individual programming task. **(4 marks)**

b) Give TWO examples of good coding practice and techniques so that program code can be readily shared amongst other programmers in the team. **(4 marks)**

c) List FOUR of the key documents that the Project Manager is responsible for once the software project has been completed and is ready to hand over to the client. **(4 marks)**

B8. Establishing a specification of *the functional and non-functional requirements* of a new software product is a key stage during the Software Development Life Cycle (SDLC).

a) Briefly compare and contrast functional and non-functional requirements. **(4 marks)**

b) Briefly explain the techniques a software developer would use to capture functional requirements from a client. **(4 marks)**

c) An important outcome of this stage is to present a specification to the end user who is also the customer/client. You may assume that the end user is fairly computer literate but will not understand technical language.

Explain how this would be presented to a customer so that it is understandable, and the customer can confirm that what is being proposed is acceptable.

(4 marks)

B9.

a) List SIX design principles that assist end user interaction with a Graphical User Interface (GUI). **(6 marks)**

b) A “form” is a type of GUI to assist in user interaction.

Produce a diagram or sketch showing the components and structure of a form-based GUI that supports user interaction for **ONE** of the following example applications:

- i) Register users to enable access to services;
- ii) Shopping for goods/services;
- iii) Viewing and booking holidays or vacations.

(6 marks)

[Turn Over]

B10.

a) What is the difference between open source and closed source software? Give an example of both open and closed source software packages covering the same application area.

(6 marks)

b) Explain what factors are considered when choosing between an open source software product/package and a closed source software product/package.

(6 marks)

B11. Explain using examples the differences between **EACH** of the following pairs of software concepts:

a) Compiler and an Interpreter;

(4 marks)

b) Indexed and sequential file access;

(4 marks)

c) Stack and a Linked List as storage structures.

(4 marks)

B12.

a) Describe the overall functions of the following types of software testing.

i) Acceptance testing;

ii) Alpha and Beta testing.

(8 marks)

b) Briefly describe the types of software testing that should be undertaken prior to acceptance testing.

(4 marks)

End of Exam