BCS THE CHARTERED INSTITUTE FOR IT

BCS HIGHER EDUCATION QUALIFICATIONS
BCS Level 4 Certificate in IT

SOFTWARE DEVELOPMENT

Thursday 17th April 2025 - 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 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.

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

A1.

a) Explain the concept of **each** of the below types of file operations. For each operation, describe the process by which the programmer would implement the operation. You do **not** need to present any code samples.

   i. Creating a file.
   (2 marks)

   ii. Storing content in a file.
   (2 marks)

   iii. Retrieving content from a file.
   (2 marks)

b) The following are three types of files. Define **each** of these file types and describe a scenario where that specific type of file would be most appropriate.

   i. Sequential.
   (4 marks)

   ii. Index-sequential.
   (4 marks)

   iii. Random access.
   (4 marks)

c) The following are two types of file formats. Explain **each** of the file formats and discuss the advantages and disadvantages of using these file formats for storing and exchanging data.

   i. Text files.
   (4 marks)

   ii. Semi-formatted files, such as Comma Separated Value (CSV) files.
   (4 marks)

d) Provide a step-by-step explanation of how you would read data from a text file in a programming language of your choice.
(4 marks)

A2.

a) The following are techniques for checking code. For **each** technique, provide a definition and a discussion of how the technique should be applied during the software development life cycle:

i. Dry-running.

(3 marks)

ii. White-box testing.

(3 marks)

iii. Black-box testing.

(3 marks)

b) You have been asked to write a test case specification for a software component, such as a function or method.

i. Explain what is meant by the term test case specification and state its purpose.

(2 marks)

ii. Describe the typical contents of a test case specification.

(6 marks)

iii. Provide a simple example of a test case specification for a function to calculate the area of a rectangle.

(4 marks)

c)

i. Define the concept of Quality Assurance (QA) in the context of software development and testing.

(3 marks)

ii. Identify QA techniques intended to help meet **three** of the following challenges in software development.

- Preventing Defects.
- Ensuring Compliance.
- Enhancing Collaboration.
- Establishing Metrics for Quality.
- Optimizing Testing Processes.
- Enhancing Customer Satisfaction.
- Continuous Improvement.

(6 marks)

[Turn Over]

A3.

a)

i. Define the concept of a data structure and explain how you would select an appropriate data structure during software development.

(2 marks)

ii. Describe the following types of data structure providing examples of **each**:

- Linear data structures.
- Non-linear data structures.

(8 marks)

b) Describe the key features of **each** of the following data structures and discuss their strengths and weaknesses in terms of storage, retrieval, and modification of data.

i. Arrays.

(4 marks)

ii. Lists.

(4 marks)

iii. Tuples.

(4 marks)

c) Compare and contrast the following abstract data types. Explain their fundamental characteristics and the way they are typically used in programming.

i. Queues.

(4 marks)

ii. Stacks.

(4 marks)

A4.

a)

i. Define the following data types giving **two** examples of **each**:

- Elementary / primitive.
- Derived.

(6 marks)

ii. Using an example in a programming language of your choice, explain how derived types are constructed from elementary types.

(4 marks)

b)

i. Explain how subtypes are used in object-oriented programming.

(4 marks)

ii. Provide an illustrative example of a situation where subtypes would be beneficial.

(4 marks)

c) Explain how strong typing in a programming language contributes to **each** of the following. Illustrate your answer with suitable examples of code written in a language of your choice:

i. Program reliability.

(4 marks)

ii. Code readability.

(4 marks)

iii. Code maintenance.

(4 marks)

[Turn Over]

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

B5.
This question is about a simple algorithm for a hash function and using it to store numbers in a hash table (an array).

The hash function, called fnHash, takes an integer (i) and divides the integer by 5. The remainder (R) is the hash key.

For example, if the integer (i) is 11, then 11 divided by 5 gives a remainder of 1 which is the hash key for the integer 11.

Thus, the hash key (R) = 11 MOD 5 = 1, where MOD is the modulus division operator.

The hash code (R) is then used as an index to the hash table, and the integer (i) is stored at row R, column 0 in the table. Fig B5 below shows the hash table with the value 11 stored at row 1 in column 0.

The hash table is represented by an array with 5 rows and 10 columns. Apart from the cell x=1, y=0 all other elements are marked as "empty" until they are populated with values.

Fig B5 Hash Table:

← Column (y) →

Row (x)	0	1	2	3	4	…	9
0							
1	11						
2							
3							
4							

A collision occurs when two integers hash to the same key value. If this occurs, then the integer is placed in the next available space on the same row. For example, the number 16 will also have the hash key of 1 and would be placed in cell x=1, y=1.

a) Redraw the hash table (Fig B5) after using the hashing function on the following integers:

23, 5, 8, 17

(4 marks)

b) Using a programming language of your choice, write the function *fnHash(i)* that returns the hash key for a given integer i.

(2 marks)

c) Write a program fragment that calls the function *fnHash(i)* and places the resultant hash key in the correct position in the hash table. You can assume that the hash table **hashTable(5,9)** has been declared and all elements initialised to "empty".

(6 marks)

B6.

a) Explain the relationship between debugging and testing program code.

(4 marks)

b) During debugging, developers often take a step-by-step approach.

Assume you have been hired to debug a piece of faulty program code, but you have not been involved in the development of the code/application.

List and explain the steps required to debug the faulty program code. There are about **four** steps in total.

(8 marks)

B7.

A website that has a UI with a good navigation scheme allows visitors to find content and features on a site.

Describe **four** basic principles that contribute to a good navigation scheme for a website.

(12 marks)

B8.

Explain the following software testing techniques:

i. Functional testing.

(3 marks)

ii. Static code testing and analysis.

(3 marks)

iii. Stress testing.

(3 marks)

iv. Load testing.

(3 marks)

B9.

a) Explain **four** benefits of using functions in a program.

(8 marks)

b) Explain what is meant by “byte code” used in Java and other languages like Python.

(4 marks)

[Turn Over]

B10.
In an experiment, 80 measurements are recorded in an array. The first 4 are shown in the table below:

Measured values								
Measurement number	1	2	3	4	5	6	7	80
Actual value	96	45	94	81	….	….	….	…

The expected values for this experiment are shown in the following table:

Expected (control) values:								
Measurement number	1	2	3	4	5	6	7	80
Expected value	100	50	75	80	….	….	….	…

The requirement is to determine the accuracy of the measured data by comparing the measured value with the expected value using a scoring system as follows:

- Score = 2 If the difference is not more than 5%
- Score = 1 If the difference is between 5-10%
- Otherwise Score = 0

For example:

- For measurement 1: the difference is 100-96=4, which is 4%. Since this is less than 5% the score recorded = 2
- For measurement 2: the difference is 50-46 =4, which is 8%. Since this is between 5 & 10 % the score recorder = 1

The following table shows the corresponding first four computed scores.

Measurement number	1	2	3	4
Score	2	1	0	2

The total score is calculated for the data set and the result printed out.

a) Draw a flow chart to calculate all 80 of the scores and print the total.

(8 marks)

b) Discuss the pros and cons of storing the measured values and the calculated scores in a csv (comma separated value) file.

(4 marks)

B11.
Software typically undergoes changes and has many versions over its lifetime.

a) Provide **six** reasons why it is important to document changes made to different versions of a program or software product over its lifetime.

(6 marks)

b) Compare and contrast **internal** software documentation (for programmers) and **external** software documentation (for end users) given the above context.

(6 marks)

B12.
You have been asked to provide a program to check that a password meets certain conditions before it is accepted.

The conditions are as follows:

- The password is a string of at least **five** characters.
- At least **one** character must be a number.

A character represents either letter (A to Z) or (a to z) or number (0 to 9).

Draw a flowchart that inputs a password as a string and tests whether the password meets the requirements above.

The following string functions are provided for you.

- IsNumeric() returns TRUE if character is a number.
- IsLetter() returns TRUE if character is a letter.
- Len(string) returns the length of the password.
- Midchar(string, position) returns the character in the specified position from the start.

(12 marks)

END OF EXAMINATION