GCE AS/A LEVEL

S25-2500U20-1

2500U20-1

MONDAY, 19 MAY 2025 – AFTERNOON

COMPUTER SCIENCE – AS unit 2
Practical Programming to Solve Problems

2 hours

The decryption password for the encrypted media files is checks-186f4

2500U201
01

INSTRUCTIONS TO CANDIDATES

The question paper is divided into two main sections. Below are the recommended timings for this assessment.

Section A

Answer **ALL** questions in **Section A**.

You are advised to spend no more than **60 minutes** on this section.

You will be required to analyse and deconstruct the scenario to consider its component parts in terms that can be addressed through automated computation.

Section B

You are advised to spend no more than **60 minutes** on this section.

You will be required to develop programs that solve computing problems referring to the initial scenario.

INFORMATION FOR CANDIDATES

The number of marks is given in brackets at the end of each question or part-question.

The total number of marks available is 60.

Assessment will take into account the quality of written communication used in your answers.

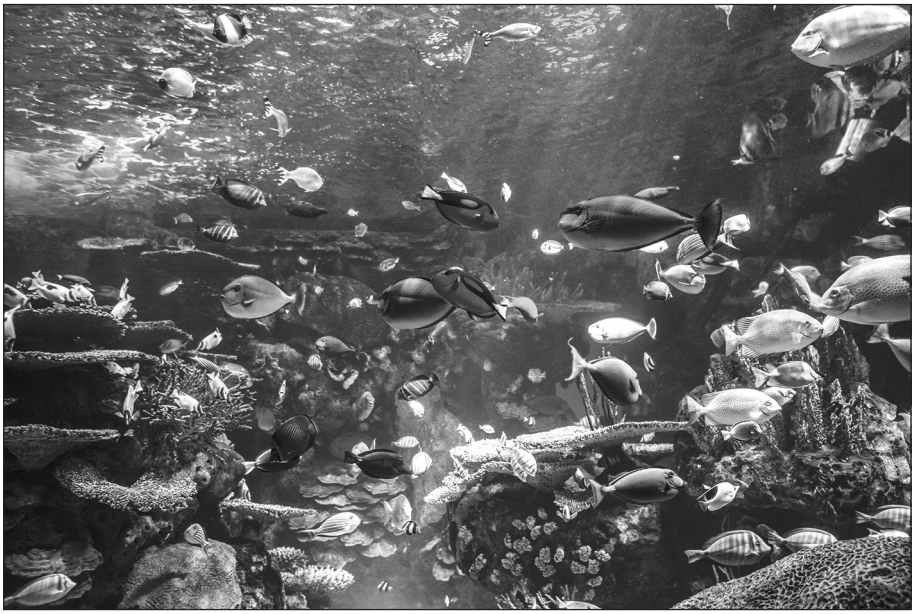
You will need a computer with an installed functional copy of the Integrated Development Environment (IDE) appropriate to your chosen programming language and word processing software.

A calculator is allowed in this examination.

Remember to save your work regularly.

No certificate will be awarded to a candidate detected in any unfair practice during the examination.

Scenario



AquaticZone Tropical Fish Sales (AZT) is a small, independent tropical fish shop that sells fish and supporting equipment to the public. The company maintains stock of tropical fish, which have to be stored carefully together in certain combinations and at certain temperatures.

The owner of **AZT** has decided to commission a new computerised system to store the information on stock such as breedID, breed name, temperature (cold/warm water) and water compatibility (freshwater/saltwater).

The main relationships of the system are:

- one breed of fish can be in many tanks
- one tank can have many breeds of fish.

You have been commissioned to develop a prototype computer system.

Section A: Analysis and Design

1. **AZT** would like to create an entity relationship diagram for the main relationships in the system. **AZT** knows there is a problem with its current relationship model as stated in the scenario.

Create a normalised entity relationship diagram for **AZT's** system. There is no need to add any attributes. [8]

2. **AZT's** owner was previously an IT professional and is concerned that moving from the existing paper-based system to a computer-based system may impact on the efficiency of his staff. The owner is considering two types of user interface: a graphical user interface (GUI) and a command-line interface (CLI).

Discuss the benefits and drawbacks of these **two** types of interface. [8]

3. The owner is considering either a phased changeover or a direct ("big bang") changeover.

Describe the advantages and disadvantages of both of these types of changeover methods.

Your answer should be related to the impact of these methods on the owner and the staff. [8]

4. (a) **AZT** is aware that only a certain amount of salt can be added to each tank to maintain the salinity level of the water. To model the potential increase in salt levels of the fish tanks, **AZT** has the following algorithm that attempts to model the increase.

```
1   Begin Subroutine salt()
2       declare a as integer
3       declare b as integer
4       declare time as integer
5       declare targetSalt as integer
6       declare iteration as integer
7
8       set time = 0
9       set a = 1
10      set b = 1
11      set iteration = 1
12
13      repeat
14
15          output iteration
16
17          set time = a
18          output time
19
20          targetSalt = a + b
21          output targetSalt
22
23          set a = b
24          set b = targetSalt
25          set iteration = iteration + 1
26
27      until iteration = 5 {next loop}
28
29  End Subroutine
```

Copy and complete the table to show the outputs of the algorithm. [8]

iteration	time	targetSalt
1		
2		
3		
4		

(b) In another algorithm, the `finalSalt` score is calculated by dividing the `targetSalt` by three and stored to **one decimal place**. The division of two of the input numbers may lead to a result that has to be truncated.

Using this new method and giving appropriate worked examples, demonstrate the effect of truncation upon the accuracy of the `finalSalt` score. [4]

5. Using a recognised convention, design an algorithm to help **AZT** control the amount of salt (salinity) within one of its tanks.

The algorithm should repeatedly accept an input called `currentLevel`, the current salinity within the tank, which is a real number.

The algorithm should attempt to maintain the level of salt in the tank between 2 and 4.

- When the salinity rises above 4, the algorithm should output “stop addition”.
- When the salinity rises above 6, the algorithm should output “add water”.
- When the salinity falls below 2, the algorithm should output “add salt”.
- When the salinity falls below 1, the algorithm should output “double the salt addition”.

The algorithm should output the current salinity and average salinity each time a reading is taken.

If the salinity is above 10 or below 0, the algorithm should output a message warning **AZT’s** owner that the salinity input is outside acceptable limits and should terminate running. [8]

Section B: Develop programs

Using the programming language that you have studied from Visual Basic, Python or Java, answer all questions within Section B.

1. (a) Open the existing program **AZStock**. This program contains code that attempts to save details on fish stock, including:

- inputting stock details
- storing stock details on disk in a text file called `stockDetails.txt`
- recalling stock details.

This code has missing/broken parts.

Complete this code.

Save your completed work as AZStock.

(b) Using the internal facility of your chosen language, add annotation to your code from part (a) that clearly explains the design of your program to another software developer. [4]

2. Create a new form or program that will allow **AZT** to implement the algorithm you designed as part of **Question 5** from **Section A**:

- Input a reading.
- If the value is 100 the message displayed should be “program terminated”.
- Output the appropriate message for the salinity input as shown in the table:

Salinity range	Message
Above 6	add water
From 4 to 6	stop addition
From 2 to 3	add salt
From 0 to 1	double the salt addition
Below 0 or above 10	fault detected

- Store the last salinity reading on disk in a text file called `saltResult.txt`
- Confirm storage of the last entry to disk on screen.
- Retrieve the last reading from disk.

Save your work as AZSalt. [12]

END OF PAPER

BLANK PAGE