



GCE AS/A LEVEL

2500U20-1



Z22-2500U20-1

FRIDAY, 27 MAY 2022 – MORNING

COMPUTER SCIENCE – AS unit 2
Practical Programming to Solve Problems

2 hours

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

Select the programming language of your choice from section **BI**, **BII** or **BIII** and answer **all** questions within your chosen section.

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.

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.

Scenario

Fred's Fish and Chip Shop



Fred's Fish and Chip Shop (Fred's) is a popular fast food shop serving freshly cooked fish and chips. **Fred's** is open seven days a week from 11am until 11pm.

Fred's needs many teams of staff to cover its long opening hours. The business currently records its data on paper and now wants a computerised system to store details of staff and stock.

The staffing structure of **Fred's** includes managers; each manager can be responsible for several teams of employees; each employee belongs to one team only. A team consists of several employees and is managed by one manager.

The main requirements of the new computer system for **Fred's** are to:

- store details on managers, teams and employees;
- store stock details including expiry dates;
- carry out simple calculations such as calculating the price of stock;
- accept electronic payment.

Section A: Analysis and Design

Answer all questions (1 – 4) in this section.

You have been asked to analyse the scenario as a preliminary step towards creating a prototype computer system for **Fred's**.

Present your answers as a single word-processed document named: **Section A**

1. (a) Create an entity relationship diagram for the **staffing structure** described in the scenario. There is no need to include any attributes. [5]

(b) Create a data structure table, including suitable validation checks, for **one** of the tables in your entity relationship diagram. [10]
2. Select and fully justify your proposed method of solution for the four main requirements listed in the scenario. [6]
3. Prepare a class diagram for electronic payments.
The class diagram should comprise:
 - One superclass with the name `Transaction`. This is to include two public properties of type integer called `accountNumber` and `transactionID`, and two public methods which each return an integer called `getAccountNumber` and `getTransactionID`
 - A subclass called `Credit` with one private property called `amount`, of type real and one private method called `setAmount` which accepts a real parameter. [7]

4. (a) **Fred's** uses this algorithm to calculate the Stock Cost based on the Cost of Purchase.

```

1  stockCost[4,2] is real {array to use}
2
3  costOfPurchase is real
4
5  input costOfPurchase
6
7  for i = 0 to 3
8    stockCost [i,0] = i + 1
9    stockCost [i,1] = costOfPurchase * ((i+1) * 1.1)
10
11 next i
12
13 End

```

Using this algorithm, copy and complete the table below showing the contents of the array `stockCost` for the input provided. [8]

`costOfPurchase = 4.0`

stockCost [i,0]				
stockCost [i,1]				

(b) This is an algorithm for a binary search.

```

1  start is integer
2  . . .
3  . . .
4  mid is integer
5
6  set start = 0
7  set endv = 9999
8  . . .
9
10 input searchValue
11
12 repeat
13     set mid = (start + endv) DIV 2
14     . . .
15     set found = TRUE
16     output "SearchValue found at position", Mid
17     endif
18
19     . . .
20     . . .
21     endif
22
23     if searchValue < stockCost[mid] then
24         set endv = mid - 1
25     endif
26     . . .
27
28     . . .
29     output "searchValue not found"
30     endif
31 End

```

Fred's intends to use the binary search algorithm shown to search the array `stockCost`. Several lines are missing from the algorithm.

Copy the algorithm into your Section A answer document and complete it using lines selected from this list; [8]

- `endv` is integer
- `set start = mid + 1`
- `if found = FALSE`
- `found` is Boolean
- `set found = FALSE`
- `if searchValue = stockCost[mid] then`
- `until (found = TRUE) OR (endv < start)`
- `if searchValue > stockCost[mid] then`

Section B: Develop programs

Select the programming language of your choice from section BI, BII or BIII and answer both questions within your chosen section.

BI Visual Basic

Fred's wants a prototype computer system to be developed using Visual Basic.

1. Open the file **stock**
 - Read through the code and familiarise yourself with its contents.
 - The file contains incomplete code which attempts to save stock details and return the number of stock items on file that match the search criteria entered e.g. expiry date.

Complete this code.

Save the changes made to the file stock. [8]

2. Using the internal facility of your chosen language, **annotate** the code from Question B1 above to clearly explain the design of this program to another software developer.

Save the changes made to the file stock. [8]

BII Java

Fred's wants a prototype computer system to be developed using Java.

1. Open the file **stock**
 - Read through the code and familiarise yourself with its contents.
 - The file contains incomplete code which attempts to save stock details and return the number of stock items on file that match the search criteria entered e.g. expiry date.

Complete this code.

Save the changes made to the file stock. [8]

2. Using the internal facility of your chosen language, **annotate** the code from Question B1 above to clearly explain the design of this program to another software developer.

Save the changes made to the file stock. [8]

Bill Python

Fred's wants a prototype computer system to be developed using Python.

1. Open the file **stock**
 - Read through the code and familiarise yourself with its contents.
 - The file contains incomplete code which attempts to save stock details and return the number of stock items on file that match the search criteria entered e.g. expiry date.

Complete this code.

Save the changes made to the file stock. [8]

2. Using the internal facility of your chosen language, **annotate** the code from Question B1 above to clearly explain the design of this program to another software developer.

Save the changes made to the file stock. [8]

END OF PAPER