

- 1 (a) Describe **two** reasons for layering in the TCP/IP model. [2]
- (b) State the main function of the Transport Layer. Describe **two** advantages of TCP over UDP. [3]
- 2 Explain the purpose and give **one** example for each of the following techniques:
- (a) Data Validation [2]
- (b) Data Verification [2]
- 3 Consider the following pseudocode algorithm:
- ```
01 FUNCTION P (n: INTEGER) RETURNS INTEGER
02 IF n <= 0
03 THEN
04 RETURN 0
05 ELSE
06 m ← 10 * P (n DIV 2) + (n MOD 2)
07 RETURN m
08 ENDIF
09 ENDFUNCTION
```
- Note that  $X \text{ MOD } Y$  gives the remainder and  $X \text{ DIV } Y$  gives the quotient when  $X$  is divided by  $Y$ .
- (a) Draw a trace diagram for the recursive function call  $P(12)$  and give the final result returned. [4]
- (b) State the purpose of the function  $P$ . [1]
- (c) State **two** key characteristics of a recursive function, and when is it suitable to be used. [3]
- (d) Give **one** advantages and **one** disadvantages of using recursive function. [2]
- 4 (a) Describe **one** advantage and **one** disadvantage of client-server network compared to peer-to-peer network. [2]
- (b) Explain the term Domain Name Server (DNS) and how it works. [3]
- (c) The school's web server provides web services to the clients and it is connected to an internal database.
- (i) Explain how requests arriving at this server are handled. [2]
- (ii) Describe **two** codes of conduct that you would expect for the IT staff. [2]
- (iii) Give **one** example of Multi-Factor Authentication to verify the identity of the IT staff. [1]

5. Winter Veterinary Care stores data in a file about pets, their owners, and appointments made and requests your help to design a relational database with the following rule:
- Each pet belongs to one owner.

The table shows the data about the appointment.

| Pet Name | Pet Breed  | Owner Name | Owner Telephone | Appointment Date | Appointment Time |
|----------|------------|------------|-----------------|------------------|------------------|
| Baguette | Dachshund  | Smith      | 62568892        | 20-03-2022       | 09:00            |
| Aster    | Chihuahua  | Roberts    | 65546741        | 20-03-2022       | 15:00            |
| Gronk    | Pomeranian | James      | 65442781        | 21-03-2022       | 09:00            |
| Biggie   | Chihuahua  | Harris     | 64432091        | 21-03-2022       | 11:00            |
| Joy      | Corgi      | John       | 66432884        | 21-03-2022       | 15:00            |
| Pooch    | Poodle     | Charles    | 62789431        | 22-03-2022       | 09:00            |
| Yoda     | Bulldog    | Patel      | 64519768        | 22-03-2022       | 09:00            |
| Hope     | Pomeranian | Harris     | 64432091        | 22-03-2022       | 15:00            |
| Pooch    | Poodle     | Charles    | 627894311       | 22-03-2022       | 17:00            |
| Yoda     | Bulldog    | Patel      | 64519768        | 23-03-2022       | 16:00            |
| Gronk    | Pomeranian | Jame       | 65442781        | 23-03-2022       | 17:00            |

- (a) Describe, with example, **two** possible issues with the way the appointments are stored. [2]

- (b) Explain whether the above table is in first normal form (1NF). [1]

Based on the information given above, design the database that consists of a number of tables.

- (c) Draw the Entity-Relationship (E-R) diagram to show the tables in third normal form (3NF) and their relationships between them. [4]

A table description can be expressed as:

TableName (Attribute1, Attribute2, Attribute3, ...)

The primary key is indicated by underlining one or more attributes. Foreign keys are indicated by using a dashed underline.

- (d) Using the information given, write table descriptions for the other tables you identified in part (c). [5]
- (e) Write an SQL query to output the pet's name, pet's breed, owner's name, Owner's Telephone and Appointment time for 22-03-2022, in ascending order of the appointment time, based on the table descriptions for the tables in part (d). [5]

Winter Veterinary Care needs to collect data for the appointments.

- (f) State **one** information that are in the scope of personal data in Personal Data Protection Act (PDPA). [1]
- (g) State **two** actions that need to be done regarding the collection, disclosure and use of this data under the Personal Data Protection Act. [2]

## 6 [VJC2022/Promo/Q2]

In a role-playing game designed using object-oriented programming, every character in the plot has the following data recorded:

| Attributes   | Description                   |
|--------------|-------------------------------|
| Combat skill | An integer, initialised as 10 |
| Endurance    | An integer, initialised as 20 |

| Properties     | Description                                                                       |
|----------------|-----------------------------------------------------------------------------------|
| Backpack Items | A suitable data structure containing items contained in the character's backpack. |
| Weapons        | A suitable data structure containing weapons equipped by the character.           |

The characters involved in the game plot are either allies or enemies.

The allies will have the following additional data recorded.

| Properties    | Description                                                                 |
|---------------|-----------------------------------------------------------------------------|
| Special Items | A suitable data structure containing special items collected by the player. |
| Disciplines   | A suitable data structure containing disciplines gained by the player.      |

In each combat encounter between the allies and enemies, the outcome is determined by comparing their combat skill and endurance scores.

The combat skill and endurance scores of an ally character are computed as follows:

| Attributes   | Description                                                                                                                                                                                                                                                    |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Combat skill | <p>Base Score = 10 + a random integer between 0 and 9</p> <p>Net score =</p> <ul style="list-style-type: none"> <li>Base score +1 for each weapon possessed by the ally</li> </ul> <p>Base score +2 if the ally possesses the "Psychic Defence" discipline</p> |
| Endurance    | Base score = 20 + a randomly generated integer from 0 to 9.                                                                                                                                                                                                    |

The combat skill and endurance scores of an enemy character are computed as follows:

| Attributes   | Description                                                                                |
|--------------|--------------------------------------------------------------------------------------------|
| Combat skill | Base Score = 10<br><br>Net score =<br>Base score +1 for each weapon possessed by the enemy |
| Endurance    | Equals own combat skill                                                                    |

(a) Draw a class diagram that shows the following for the game described above:

- the superclass
- any subclasses
- inheritance
- properties
- other appropriate methods

[12]

(b) Using an example from the game design, explain the role of:

(i) inheritance.

[3]

(ii) polymorphism.

[3]

(c) Explain why object-oriented programming is preferred over procedural programming in this game design.

[2]

## 7 [VJC2022/Promo/Q2]

Joseph is taking a toy apart. Each time he removes an item from the toy, he writes the name of the item at the bottom of a paper list. When he rebuilds the toy, he puts the items back together working from the bottom of the list.

Joseph decides to write a computer program to replace this list by using a stack, `StackContents`.

(a) Explain the concept of a stack. [2]

The stack is represented as an array in the program. The first element in the array is `[0]`. The current contents of the stack, `StackContents`, and its pointer, `StackPointer` are shown.

|                     |   |                      |
|---------------------|---|----------------------|
| <b>StackPointer</b> | 5 |                      |
|                     |   | <b>StackContents</b> |
| 0                   |   | "Screw 1"            |
| 1                   |   | "Screw 2"            |
| 2                   |   | "Back case"          |
| 3                   |   | "Screw 3"            |
| 4                   |   | "Engine outer"       |
| 5                   |   |                      |
| 6                   |   |                      |
| 7                   |   |                      |

(b) Describe the purpose of the variable `StackPointer`. [1]

The procedure `POP()` removes an item from `StackContents`. The procedure `PUSH(<identifier>)` adds an item to `StackContents`.

- (c) The current contents of the stack, `StackContents` and its pointer, `StackPointer` are shown in part (a). Copy the table below to show the contents of `StackContents` and the value of `StackPointer` after the following code is run.

```
POP()
POP()
PUSH("Light 1")
PUSH("Light 2")
PUSH("Wheel 1")
POP()
POP()
```

|                     |                      |                      |
|---------------------|----------------------|----------------------|
| <b>StackPointer</b> | <input type="text"/> | <b>StackContents</b> |
| 0                   |                      |                      |
| 1                   |                      |                      |
| 2                   |                      |                      |
| 3                   |                      |                      |
| 4                   |                      |                      |
| 5                   |                      |                      |
| 6                   |                      |                      |
| 7                   |                      |                      |

[2]

`StackContents` is implemented using a 1-dimensional array, declared as:

```
DECLARE StackContents : ARRAY[0:7] OF STRING
```

The procedure `POP` outputs the last element that has been pushed onto the stack and replaces it with a `'*'`.

- (d) Write the pseudocode for the procedure `POP`. [5]

A linked list is a collection of nodes, each containing a pointer pointing to the next node.

- (e) Describe how memory is allocated for a linked list. [2]

- (f) Explain how the stack, `StackContents` can be implemented using a linked list. [3]

8. Pythagoras investigated the sum of proper divisors in his study of numbers, this is known as the Aliquot sum. For example, the Aliquot sum of 6 is 6 (given by  $1+2+3$ ) since 1, 2 and 3 are its proper divisors, while the divisor sum of 8 is 7 ( $1+2+4$ ).

- (a) Write a function `d_sum(number)`, using **pseudocode**, which takes in an integer `number` and returns the Aliquot sum of `number`. [3]

We say that a number is perfect if its Aliquot sum is equal to itself. Hence, 6 is a perfect number while 8 is not perfect.

- (b) Write a **pseudocode** statement, using `d_sum(number)` to determine whether `number` is perfect. [1]

- (c) Write a **pseudocode** procedure which outputs the first 20 perfect numbers. [3]

We say that  $(num1, num2)$  is an amicable pair if their Aliquot sum is equal to each other. For example,  $(220, 284)$  is an amicable pair since `d_sum(220) = 284` and `d_sum(284) = 220`.

- (d) Prove that if  $(num1, num2)$  is an amicable pair, then

$$d\_sum(d\_sum(num1)) = num1. \quad [1]$$

- (e) Determine the value of `d_sum(d_sum(num1))` if `num1` is a perfect number. [2]

It is clarified that a number cannot be an amicable pair with itself. (i.e.  $num1 \neq num2$ .)

- (f) Using the function in part (a) and statement in part (d), write a **pseudocode** procedure, which finds all amicable pairs less than 10000. [4]

9. Write your code in **Python** for this question.

(a) A list `Sequence` contains  $n$  non-zero integers. Write a program to generate another list `Product` where the  $i$ -th element in `Product` is obtained by multiplying the first  $i$  integers in `Sequence`. For example, a list `Sequence` = [1, 3, 5, 7, 9] will generate `Product` = [1, 3, 15, 105, 945]. [4]

(b) The Conway's Game of life is built on a 5x5 grid of cells, which are in one of two possible states, live or dead. Every cell interacts with its neighbours, which are the cells that are horizontally, vertically, or diagonally adjacent. For example, the cells A, B, C below have 8, 5 and 3 neighbours respectively.

|   |  |   |  |   |
|---|--|---|--|---|
|   |  |   |  |   |
|   |  |   |  |   |
|   |  | A |  |   |
| B |  |   |  |   |
|   |  |   |  | C |

Assume 1 and 0 represents live and dead respectively. Below is a sample grid and its corresponding counting of live neighbours for each cell.

Sample Grid:

```

0 0 1 1 1
1 1 1 0 1
0 1 0 1 0
0 1 1 1 0
1 0 0 1 1

```

Counting of Live Neighbours:

```

2 4 3 4 2
2 4 5 6 3
4 5 7 4 3
3 3 5 4 4
1 3 4 3 2

```

The grid's first transition occurs with the following rules:

1. Any live cell with two or three live neighbours survives;
2. Any dead cell with three live neighbours becomes a live cell;
3. In any other cases, the cell becomes dead.

Hence, the sample grid above after transition becomes

```

0 0 1 0 1
1 0 0 0 1
0 0 0 0 1
1 1 0 0 0
0 1 0 1 1

```

Write a program to generate a random 5x5 grid and display the grid after transition.

[11]