HWA CHONG INSTITUTION
C2 PRELIMINARY EXAMINATION 2023

COMPUTING
Higher 2 Mark Scheme

13 Sept 2023 **Paper 1 (9569 / 01)** **1400 -- 1700 hrs**

---

1. The examinations department of a school needs to store data on the examinations taken by its students. Examination subjects are classified as one of the two types, namely academic subjects and practical subjects.

   For all subjects, the data that will be stored include:
   - Subject code
   - Subject name
   - Subject type

   The assessment process for an academic subject comprises two written papers. The duration of each of these two papers will be stored.

   For the practical subject's assessment process there are no written papers but there is a single practical examination. A practical examination has a duration and a final date.

   (a) Draw a diagram that shows suitable classes and their relationships for a solution to this problem that uses OOP techniques. Include appropriate attributes and methods in each class. [6]

   (b) The school also offers vocational subjects to their students. A vocational subject has no formal examinations, but the assessment is based on the work that the students complete throughout the course. However, this coursework has to be completed by a certain date.

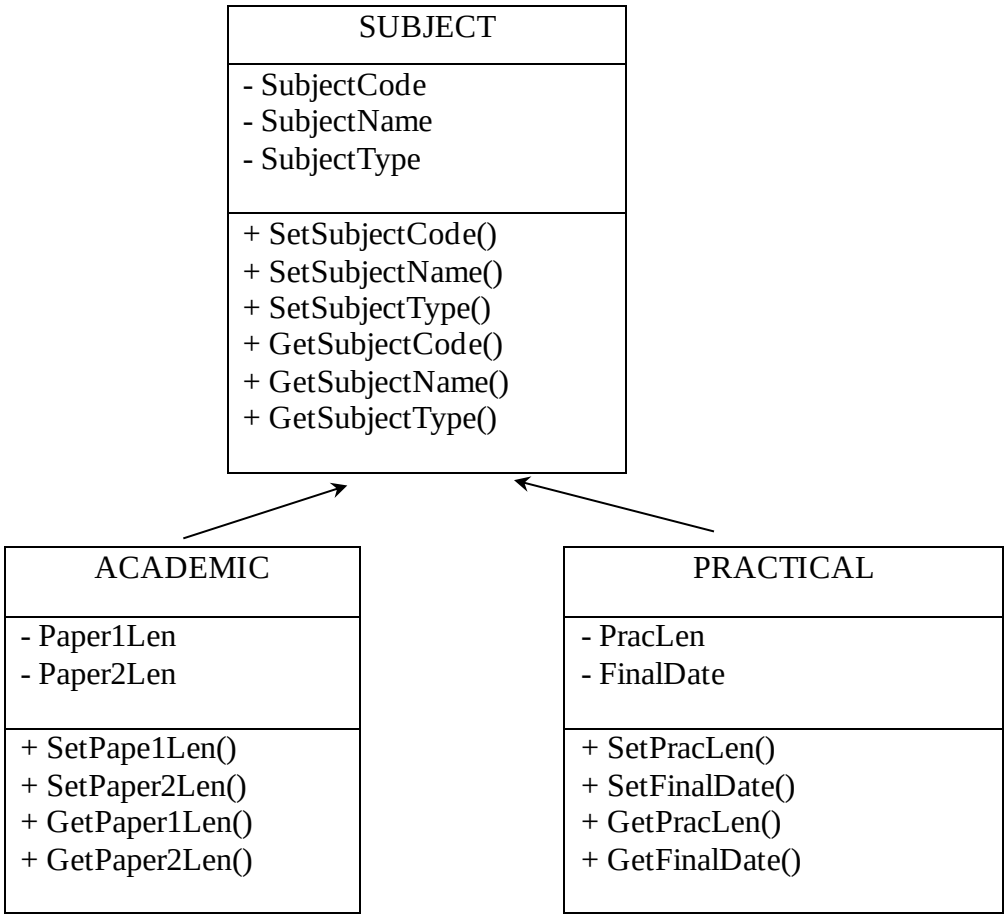
   Suggest any changes that need to be made to the classes used in **part (a)**. [3]

   (c) Explain the difference between a class and an object. [2]

   (d) Give **two** benefits of Object-Oriented programming. [2]

Solution:

(a)



Superclass (Subject) and two subclasses (Academic, Practical)
Inheritance from Subject superclass (arrows point upward)
Attributes and methods for Subject class
Attributes and methods for Academic class
Attributes and methods for Practical class
Private attributes (minus sign) and public methods (+ sign)

(b) Define a new class: Vocational
Inherits from Subject class
Attribute: CompletionDate Methods: getting and setting CompletionDate

(c) **Class:**

- User defined data type
- Used as a blueprint to create individual objects that have common attributes and methods

Object:

- Is an instance of a class

- Each object has unique values to the properties defined in the class
- Take up memory space

(d) **Benefits of OOP:**
- Reuse of code through inheritance
- Enhance security through encapsulation and abstraction
- Flexibility through polymorphism
- Modularity for easier troubleshooting
- Effective problem solving
- Methods can be changed without affecting how they are used.

Marker's Comments:
(a) Students should name classes in the singular. E.g. Subject instead of Subjects.
A few responses had the inheritance arrows pointing in the wrong direction or had no arrows.

(b) The question asks to suggest any changes to the classes, you must explain the changes clearly rather than drawing a new class diagram.

(c) **Class**: Many answers used, correctly, words such as 'template' or 'definition' but failed to make it clear that a class is a template or a definition for a collection of objects, not a single object. These objects have a common set of properties and methods.
Object: Many correctly stated that an object is an instance of a class and the object's properties would have real data.

(d) Answers that scored marks included re-use being easier (as a result of inheritance), data being hidden / protected and only accessible from public methods.

2. (a) Describe **one** advantage and **one** disadvantage of client-server network. [2]

(b) Describe **one** disadvantage of packet switching and how the problem can be handled. [2]

(c) Describe **two** differences between switch and router in the TCP/IP model. [2]

(d) Explain the term Domain Name Server (DNS) and how it works. [5]

Solution:

(a) **Advantage**: if **one client is down**, the whole system can still continue to work. OR Server has **access control** of clients.
Disadvantage: If the **server is attacked**, the whole system is down. OR Server requires a high level of **maintenance**, e.g. cost of hardware, professional training.

(b) It requires more **time for reassembly** at the destination device as packets may arrive in random order. To handle this, a **sequential number** is attached to each packet, and efficient algorithms are used to sort the packets into the original order.

(c) Switch uses **MAC addresses** at the **Data Link** layer, Router uses **IP address** at the **Network** layer.

(d) DNS server **translates domain name** in human language to **IP addresses** in machine language.

DNS server first checks if the requested domain name is available in its **cache**. If not found in its cache, it sends request to the **local DNS** by the Internet Service Provider. If still not found, DNS searches from a **hierarchy of distributed database** to locate the domain name. Once found, DNS server **sends the IP address** back to the user.

Marker's Comments:
Network and Security requires careful reading of the question (no missing answers, no misunderstanding) and content answer (no short answers with few words). Answer within the syllabus is highly recommended, marks may differ for out-of-syllabus answers.
(b) The problem can be handled within packet switching, not to change to circuit switching.
(c) Many students did not specify the layer in the model.

3. (a) Give **one** example of multi-factor authentication. [1]

   (b) Explain how a Denial of Service (DOS) attack can compromise an internal server and suggest **one** protection scheme to detect a DOS attack. [3]

   (c) Give **two** purposes of using digital signature and describe how it works. [8]

Solution:
(a) multi-factor from the three categories:
- Something you know: password
- Something you have: OTP, token, access card, passport/NRIC card
- Something you are: fingerprint, retina, voice, face

(b) DOS attacks the network traffic by **sending massive request** to exhaust server resources and bandwidth, and hence **disables the server from responding to legitimate requests**.

It can be detected by firewall, intrusion detection/protection system.
(c) purpose:
- Authentication: the message was created by the known sender
- Non-repudiation: the sender cannot deny having sent the message
- Integrity: the message is not altered in transit

Process:

- The sender uses a **hash algorithm** to create a **hashed version of the message**
- The sender uses **its private key** to encrypt the hash to the **digital signature**
- Both the message (encrypted or not) and the digital signature are **sent to the receiver**
- The receiver uses the **sender's public key** to decrypt the digital signature back to the sender's version of hash
- The receiver uses the same hash algorithm to **create a new hash** from the received message
- If the two hashes **match**, it means the data is not altered and is sent by the known sender

4. (a) The following are the inorder and postorder traversal of a binary tree whose nodes are labelled 1, 2, . . . , 9.

   Inorder: 4, 7, 2, 1, 5, 3, 8, 6, 9

   Postorder: 7, 4, 2, 5, 8, 9, 6, 3, 1

   (i) Draw the binary tree with the nodes labelled. [3]

   (ii) Give the preorder traversal of this binary tree. [2]

   (b) The following names are to be inserted in a binary search tree in the order given:

   `Lorna, Peter, Romeo, Betty, Anthony, Linda, Matthew`

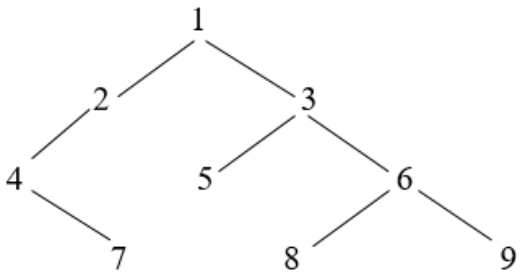
   Assuming an empty tree to start with, show how these names will be stored by drawing a tree diagram. [2]

   (c) By using suitable example and diagram, describe a situation where a binary search tree becomes inefficient. [2]

   (d) Give an advantage of using a binary search tree for storing data over a linked list. [1]

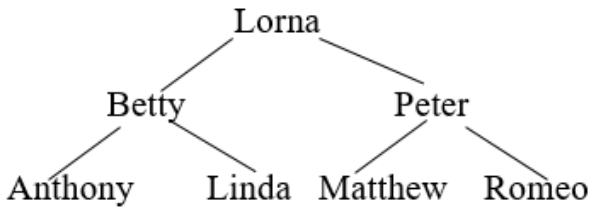
Solution:

(a)(i)



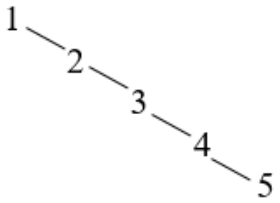
(ii) Preorder: 1, 2, 4, 7, 3, 5, 6, 8, 9

(b)



(c) (i) If the keys to be inserted are sorted, then it will produce a tree that degenerates into an inefficient chain.

e.g. Consider the list of numbers: 1, 2, 3, 4, 5 to be inserted into a BST



traversing through the BST will end up becoming a linear search

(ii) Advantages

- Better search times O(log(n)) as opposed to O(n).
- Different ways of traversing a tree mean that items can be stored in one sequence and retrieved in a different sequence.

Marker's Comments:

(a) Majority of students showed knowledge of binary tree traversal.
Error carried forward is applicable for part (ii), which means if you did get part (i) wrong, you wouldn't get marked down because of it for part (ii).

(b) Students scored well on BST construction.

(c) Many students mentioned that the tree is unbalanced, but failed to state what makes a binary tree unbalanced.

(d) May use the Big O notation to compare the time complexity of binary search (for binary search tree) and linear search (for linked list) algorithms.

5. In a country club, every member is issued a Membership Identification Number (MIN). MIN is a unique number comprises of 10 digits separated by two dashes. The last digit of MIN is a check digit using a modulus-10 system.

Consider the MIN 456-789-5468. The check digit is calculated based on the left-most 9 digits. The first step is to double every alternate digits, starting with the second-last digit and moving left. Then sum all the digits, including both the unchanged digits as well as those doubled (e.g. 18 contributes 1 + 8). The following illustrates the process:

Digit	Double	Sum of digits
6		6
4	8	(0+8)
5		5
9	18	(1+8)
8		8
7	14	(1+4)
6		6
5	10	(1+0)
4		4
	Total Sum	**52**

The total sum 52 is then divided by 10. If the remainder is 0, then the check digit is 0, otherwise, use 10 to subtract by the remainder to obtain the check digit.

(a) Using **pseudocode**, write an algorithm to validate a MIN number. [6]

(b) Using the algorithm in **part (a)**, determine whether the following MIN numbers are valid.
(i) 456-789-0126
(ii) 986-745-0124 [2]

(c) The algorithm for verifying MIN number contains a few weaknesses. Identify a weakness and explain how it can be solved. [2]

Solution:

(a)
```
dash = '-'
digitString ← ''
FOR i ← 1 TO 12            # 10 digits + 2 dashes
    IF MIN[i] <> dash
        digitString ← digitString + MIN[i]
    ENDIF
ENDFOR
sum ← 0
FOR i ← 9 TO 1 STEP -1     # from second last digit
      digit ← INTEGER (digitString[i])
      IF  i MOD 2 = 0          # even position digit
            digit ← digit * 2
            IF digit > 9
                digit ← digit - 9       # digit ← 1 + (digit MOD 10)
            ENDIF
      ENDIF
      sum ← sum + digit
 ENDIF

checkDigit ← INTEGER(digitString[10])       # last digit
totalWeightedSum ← sum + checkDigit
IF totalWeightedSum MOD 10 = 0
       OUTPUT 'Valid'
ELSE
       OUTPUT 'Invalid'
ENDIF
```

(b) (i) 456-789-0126

The total weighted sum + check digit

$= (2+2+0+9+8+5+6+1+4) + 6 = 43$, is not divisible by 10 $\Rightarrow$ Not valid

(ii) 986-745-0124

The total weighted sum + check digit

$= (2+2+0+1+4+5+6+7+9) + 4 = 40$, is divisible by 10 $\Rightarrow$ Valid

(c) - Transposition error of odd (or even) position digits will not be detected e.g 456-789-5468 if typed as 654-789-5468 (or e.g 456-789-5468 if typed as 476-589-5468) because they result in the same check digit.

- Assign different weights to the odd (or even) position digits

Marker's Comments:

(a) Must indicate the pseudocode algorithm uses 0 or 1-based indexing.
Avoid using Python syntax in written paper. e.g. `MIN[-1]`; `MIN[:-1]` and `for digit in MIN` are not considered as pseudocode. To extract character in a string with pseudocode, we will make use of the index, e.g. `MIN[i]`.
`MIN` is a 12-character string, the most common error was to omit checks for dashes, and also failed to apply the '`int`' function when performing calculation.

(b) Need to show some evidence of the method, based on the algorithm written in part (a).

(c) Transposition errors of alternate digits were common correct answers.
Many incorrect responses made reference to errors connected to data input validation such as invalid length or invalid format error. These tests would have been used to ensure that the check digit test was only applied to a valid input MIN number.

6. The school wants to deploy an efficient method to store and search students' names.
Student A suggests entering the names into an array and using binary search.

(a) Explain why the array must be sorted before performing binary search. [1]

(b) State **two** features of a successful recursive function. [2]

Names are stored in `Array` in ascending order. Student A writes the recursive function below to search for `Target` in `Array`. It returns `True` if `Target` is found and `False` otherwise.

```
01   FUNCTION B(Array, Target, Low, High) RETURNS BOOLEAN
02        IF Low > High THEN
03             RETURN False
04        ELSE
05             Mid ← (Low + High) div 2
06             IF Target < Array[Mid] THEN
07                  B(Array, Target, Low, Mid - 1)
08             ELSE IF Target > Array[Mid]
09                  B(Array, Target, Mid + 1, High)
10             ELSE
11                  RETURN True
12             ENDIF
13        ENDIF
14   ENDFUNCTION
```

Note: the `div` operation returns an integer value after division, e.g. 7 div 2 = 3.

(c) State the significance of line 02. [1]

(d) Name the type of error for lines 07 and 09. Modify these two lines to make the function work. [2]

(e) Write down the statement to call the function to perform the binary search. [1]

Student B suggests storing the names in a hash table and using hash table search.

(f) State **two** features of a good hashing algorithm. [2]

(g) Explain how two different records hashing to the same location can be managed. [2]

(h) Use time complexity to explain the advantage of a hash table search might have over a binary search. [2]

Solution:

(a) if the array is sorted (for example, in ascending order), when we compare the target to the middle element of the array, (for example, target smaller than the middle element), we can move to the lower half of the array with all the elements smaller than the target.

(b) Call itself recursively, have base case, reduce to simpler version of itself.

(c) Terminal case when target is not found

(d) Logical error.

```
07    RETURN B(Array, Target, Low, Mid – 1)
09    RETURN B(Array, Target, Mid + 1, High)
```

(e)

```
# array index starts from 0
B(Array, Target, 0, LENGTH(Array) - 1)
OR
# array index starts from 1
B(Array, Target, 1, LENGTH(Array))
```

(f)

Easy to compute: It should be efficient and easy to compute

Uniform distribution: It should provide a uniform distribution across the hash table and should not result in clustering

Less collisions: Collisions occur when pairs of elements are mapped to the same hash value. These should be avoided

(g) Chaining: Each slot of the hash table is a linked list. Store all records that hash to the same location in the same linked list

Linear Probing: Perform a circular linear search of the table from the location where the collision occurs, continue until an empty slot is found and store the record

(h) Ideally if there is no collision, hash table search time complexity if O(1). But binary Search time complexity is O(logn).

Marker's Comments:
(c) terminal case ends the recursion here. Base case determines when the recursion continues.
(d) three types of error: syntax error, logical error, run-time error. The codes can run but do not give correct output, hence it is a logical error.
(f) hashing algorithm refers to the hash function, not the hash table insertion or search algorithm.
(g) this question requires only one method, many students attempted to describe two methods but neither is described fully correct. Linear probing requires circular linear search for the next available slot. Chaining requires linked list of dynamic sizes, not array since array is fixed size but we cannot tell how many spaces to reserve for collision.
(h) the advantage of hash table search might have over binary search, we can compare ideal case of hash table. Worst-case of hash table is O(n), note that mathematically n > logn for integer n. .

7. In a football league, there are several teams. A database is created to store data about the clubs and the players that play for them.

   - Each player belongs to a single club.
   - A club can have multiple players, but only one coach.
   - Each player can play multiple positions.

   This table shows the data about the clubs and their players:

ClubID	Club Name	Club City	Coach Name	PlayerID	Player Name	Player Age	Position Code
14	Cannons	Islington	Mickey	9663	Dale	25	GK
				4336	Aliba	22	DF
				2603	Blanc	25	DF MF
				8257	Bukasa	22	DF MF FW
				1160	Marty	22	MF FW
				3984	Fan	24	DF MF
				2115	Øguard	24	MF
6	Pies	Tyne	Hao	4988	Popo	31	GK
				1846	Mares	25	DF MF
				6277	Toni	23	DF MF
				5956	Ishak	23	FW
				7555	Batman	23	DF
				9616	Wilcock	24	MF FW
12	Zombies	Salford	Rick	1849	Anono	27	GK
				7951	Ford	25	FW
				8293	McGuy	30	DF

(a) Explain whether the above table is normalised. [1]

The following table is created to keep track of the physical attributes required to play these positions, allowing the coach to better monitor the players' development.

Table: `POSITION`

PosCode	PosName	MainSkill	SecondarySkill
GK	Goalkeeper	Agility	Strength
DF	Defender	Strength	Technique
MF	Midfielder	Speed	Technique
FW	Forward	Speed	Agility

A relational database is to be used. Using the information above, design the database that consists of a number of tables.

(b) 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.

The `POSITION` table has been identified with the table descriptions:

`POSITION (Poscode, PosName, MainSkill, SecondarySkill)`

(c) Using the information given, write table descriptions for the other tables you identified in part **(b)**. [4]

The player McGuy changed his club from Zombies to the Pies.

(d) Write an SQL query to reflect this in the database. [2]

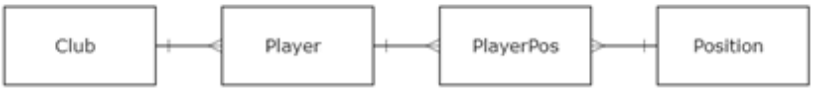
A new player joined the club Zombies. The following are his personal information:

PlayerID	Player Name	Player Age	Position Code
7777	Redondo	38	MF FW

(e) Write SQL query statement(s) to add the player into the database. [4]

(f) Write an SQL query to output the name, club name and age of all the players with 'Speed' as their main skill, ordered by their age from youngest to oldest. [6]

(g) The Zombies collected their players' health data to monitor their development and match fitness. This data was later used in advertisements for an energy drink. State the Data Protection Obligation breached under PDPA, and explain what the club should have done before using the data in advertisements. [2]

Qn	Answer
(a)	The table is not normalised as it is not atomic. A player can have multiple positions and a club can have multiple players.
(b)	
(c)	Club(ClubID, Name, City, Coach) Player(PlayerID, ClubID, Name, Age) PlayerPos(PlayerID, PosCode) or PlayerPos(PlayerPosID, PlayerID, PosCode)
(d)	UPDATE Player SET ClubID = 6 WHERE PlayerID = 8293
(e)	INSERT INTO Player (PlayerID, ClubID, Name, Age) VALUES(7777, 12, 'Redondo', 38) INSERT INTO PlayerPos (PlayerID, PosCode) VALUES(7777, 'MF') INSERT INTO PlayerPos (PlayerID, PosCode) VALUES(7777, 'FW')

(f)	SELECT Player.Name, Club.Name, Player.Age FROM Player INNER JOIN Clubs ON Player.ClubID = Club.ClubID INNER JOIN PlayerPos ON Player.PlayerID = PlayerPos.PlayerID INNER JOIN Position ON PlayerPos.PosCode = Position.PosCode WHERE Position.MainSkill = 'Speed' ORDER BY Player.Age ASC
(g)	Purpose Limitation is breached as purpose of use is not approved. The club should seek the consent of the players for their personal health data to be used for advertising.

Marker's Comments:
(b) some students have a Coach table and Club-Coach. 1-to-1 relations are usually not required.
(d) when databases are updated, the keys are the ones that MUST be changed
(e) note that Redondo's age must be integer as that is used in comparisons such as qn (f), this means that PosCode and name must be in string
(f) Question asked for MainSkill = 'Speed'. Students SHOULD follow the requirements given as the position table could be modified.
(g) Consent was previously given to use health data for development and match fitness. However, it is now used for a different purpose. While purpose limitation obligation is definitely breached, it is possible to argue that consent obligation is breached. However, we cannot just notify an individual that their data is used for other purposes without obtaining new consent.

8. The base-6 system, also known as the senary or hexary system, is a numeral system that uses six symbols to represent numbers: 0, 1, 2, 3, 4, and 5. This is a popular system due to its divisibility.

   (a) Using **pseudocode**, write a function `dec2sen(n)` that takes in a decimal number n and returns its equivalent senary (base-6) representation as a string. [5]

   (b) Give the senary-string for the decimal value of 100. [1]

   (c) Convert '1234' from senary to decimal. [1]

   The base-36 system, which uses the Arabic numerals 0-9 and the latin letters A-Z, is commonly used by URL redirection systems like TinyURL. The base-36 system is closely related to the base-6 system.

   (d) Explain why the base-36 system is closely related to the base-6 system. [1]

(a)	`FUNCTION dec2sen(n) RETURNS STRING`
	`  senary ← "  //empty string`
	`  WHILE n > 0`
	`  remainder  n MOD 6`
	`  senary ← STRING(remainder) + senary`
	`  n ← n DIV 6`
	`  ENDWHILE`
	`  RETURN senary`
	`ENDFUNCTION`

(b)	'244' (base 6, string) 2(36) + 4(6) + 4 = 100
(c)	310 (base 10, integer) 1(216) + 2(36) + 3(6) + 4 = 310
(d)	Since 36 is the 2nd power of 6, every 2 characters in the senary string can be converted to base 36 easily.

Marker's Comments:
(b) While we did not penalise, note that any number that is NOT base 10 should be a string, such as binary-STRING.
(d) Factor/multiple is not enough. 10 is a multiple of 2, and 2 a factor of 10, but they are not as closely related. Saying 6^2=36 is not enough as it does not EXPLAIN. Please write more to explain.