HWA CHONG INSTITUTION
2023 C2 TIMED PRACTICE EXERCISE

COMPUTING
Higher 2

27 June 2023 **Paper 1 (9569 / 01)** **1400 -- 1500 hrs**

READ THESE INSTRUCTIONS FIRST

An answer booklet will be provided with this question paper. You should follow the instructions on the front cover of the answer booklet. If you need additional answer paper ask the invigilator for a continuation booklet.

Answer ***ALL*** questions.

Approved calculators are allowed.

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

The total number of marks for this paper is **35**.

This document consists of 7 printed pages.

1. A printing shop offers printing services to its customers. When a printing order is sent to the shop, the following information is recorded:

- Date of order
- Name of customer
- Number of copies
- Color printing or black and white printing
- Whether express printing is required

The printing shop accepts three types of orders, namely leaflets, books and posters.

Customers printing leaflets or books need to indicate if they require single side or double side printing.

In addition, for books, the type of cover (hard cover or soft cover) would need to be recorded.

Leaflets and books are available in three paper sizes (A3, A4 or A5), while posters are only available in a fixed size of A2.

For poster printing, customers have a choice of either glossy or matte finishing.

The total charge to customer is determined by the specifications of the order and the formula is unique for each type of order.

This system is to be implemented using object-oriented programming (OOP).

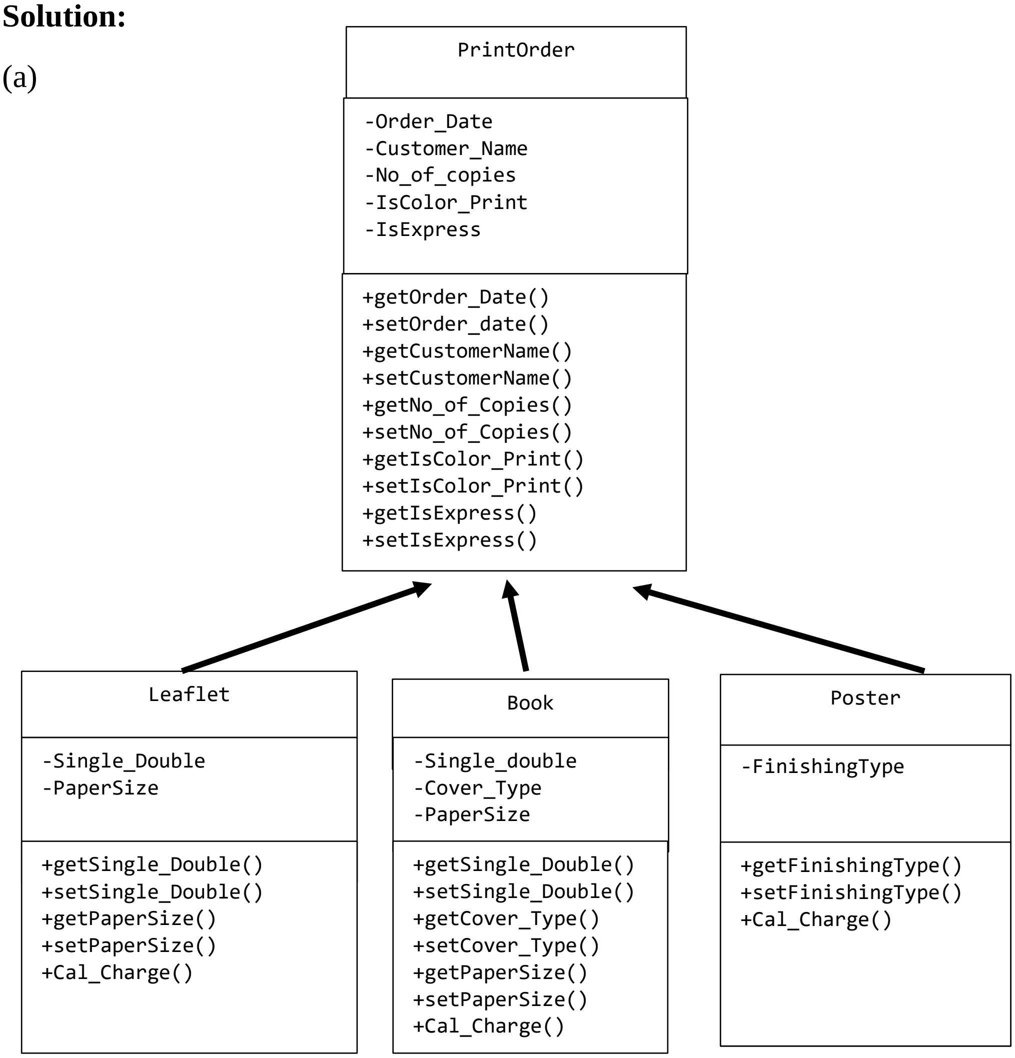
(a) Draw a class diagram that shows the following for the application described above.

- the superclass
- any subclasses
- inheritance
- properties
- appropriate methods [7]

(b) Describe how your design in **part (a)** demonstrates code reuse. [2]

Solution:

(a)



(b)

- The sub-classes (Leaflet, Book and Poster) inherit all the attributes and methods of the PrintOrder class.
- All these common attributes (Order_Date, Customer_Name …) and methods (getOrder_Date, setOrder_Date, setCustomer_Name, getCustomer_Name …) are inherited and do not need to be rewritten/recode in the 3 subclasses, hence code reused is achieved.

2. The school would like to create a social media platform that has users, posts and tags.

All users need to have an account in the platform. Each user has a name and password.

The platform allows each user to create multiple posts. Each post consists of a title, content and timestamp of the creation. The user can also add one or more tags on the post created.

A tag consists of a keyword that helps other users find the content with ease. Users can click on the tag that will take them to a page that displays all posts with this tag.

A relational database is to be used. Based on the information given, design a database that consists of a number of tables.

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

(b) A table description can be expressed as:

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

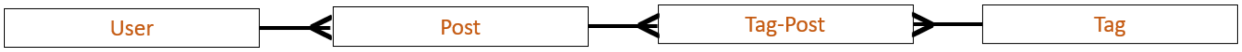
The primary key is indicated by underlining one or more attributes. Foreign keys are indicated by using a dashed underline.

Write table descriptions for the tables you identified in **part (a)**. [4]

(c) Write an SQL query to output the post's title, content, creator, timestamp of all the posts that are tagged with '#sql', in descending order of timestamp. [6]

Solution:

(a)



(b)

User(UserID, Name, Password)

Post(PostID, Title, Content, UserID, TimeStamp)

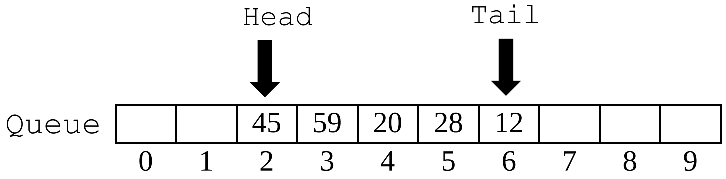
TagPost(TagID, PostID) FK for both attributes

Tag(TagID, Keyword)

(c)

```
SELECT P.Title, P.Content, U.Name, P.TimeStamp FROM
Post P INNER JOIN User U ON P.UserID = U.UserID
INNER JOIN TagPost TP ON TP.PostID = P.PostID
INNER JOIN Tag T on T.TagID = TP.TagID
WHERE T.Keyword =’#sql’
ORDER BY TimeStamp DESC
```

3. A queue data structure is implemented using an array `Queue` and two pointers, `Head` and `Tail`. The space in array is fully utilized to perform the queue operation. `Head` points to the index of the first element in the queue. `Tail` points to the index of the rear element in the queue, as shown:



Queue can either be linear or circular.

(a) Explain what is the difference between a linear queue and a circular queue. [2]

(b) Using **pseudocode**, write algorithm for the following functions for a **circular queue**:

(i) `Init()` – initializes the queue data structure. [1]
(ii) `Enqueue (data)` – adds `data` into queue data structure. [5]
(iii) `Dequeue()` – deletes data from queue data structure. [4]

Your algorithms should allow for exceptional cases.

Solution:

(a)

In a linear queue, the slots cannot be reused after the item is dequeued. In a circular queue, the slot can be re-used after an item is dequeued.

(b) (i)

```
FUNCTION Init()
     CONSTANT SIZE = 10
     Head ←  0
     Tail ← -1
ENDFUNCTION
```

(ii)

```
FUNCTION Enqueue(data)
    IF (Tail <> -1) AND
       ((Head = 0 AND Tail = SIZE-1) OR (Head = Tail+1))THEN
        OUTPUT 'Queue is Full'
    ELSE
       Tail ← Tail + 1
       IF Tail = SIZE THEN
          Tail ← 0
       ENDIF
       Queue[Tail] ← data
    ENDIF
ENDFUNCTION
```

(iii)

```
FUNCTION Dequeue()
    IF Tail = -1 THEN
       OUTPUT 'Queue is Empty'
    ELSE
       IF Head = Tail THEN
          Head ← 0
          Tail ← -1
       ELSE
          Head ← Head + 1
          IF Head = SIZE THEN
               Head ← 0
          ENDIF
       ENDIF
    ENDIF
ENDFUNCTION
```