

Theory Mock Paper 1 suggested solution

1. The Human Resource department of an organization would like to develop a system using object-oriented approach to manage the information of the employees.

One of the functions of the system is to compute the monthly pay of the full-time employees which comprise of the monthly salary and the overtime allowance.

Due to the rapid expansion of the organization, the organization starts to employ daily-rated employee. For daily-rated employee, their monthly pay are computed based on the rate per day and the number of days worked per month.

- (a) Draw a class diagram which exhibits the following:

- Suitable classes with appropriate properties and methods
- Inheritance
- Polymorphism

[6]

- (b) Explain how your design in (a) demonstrates code reuse.

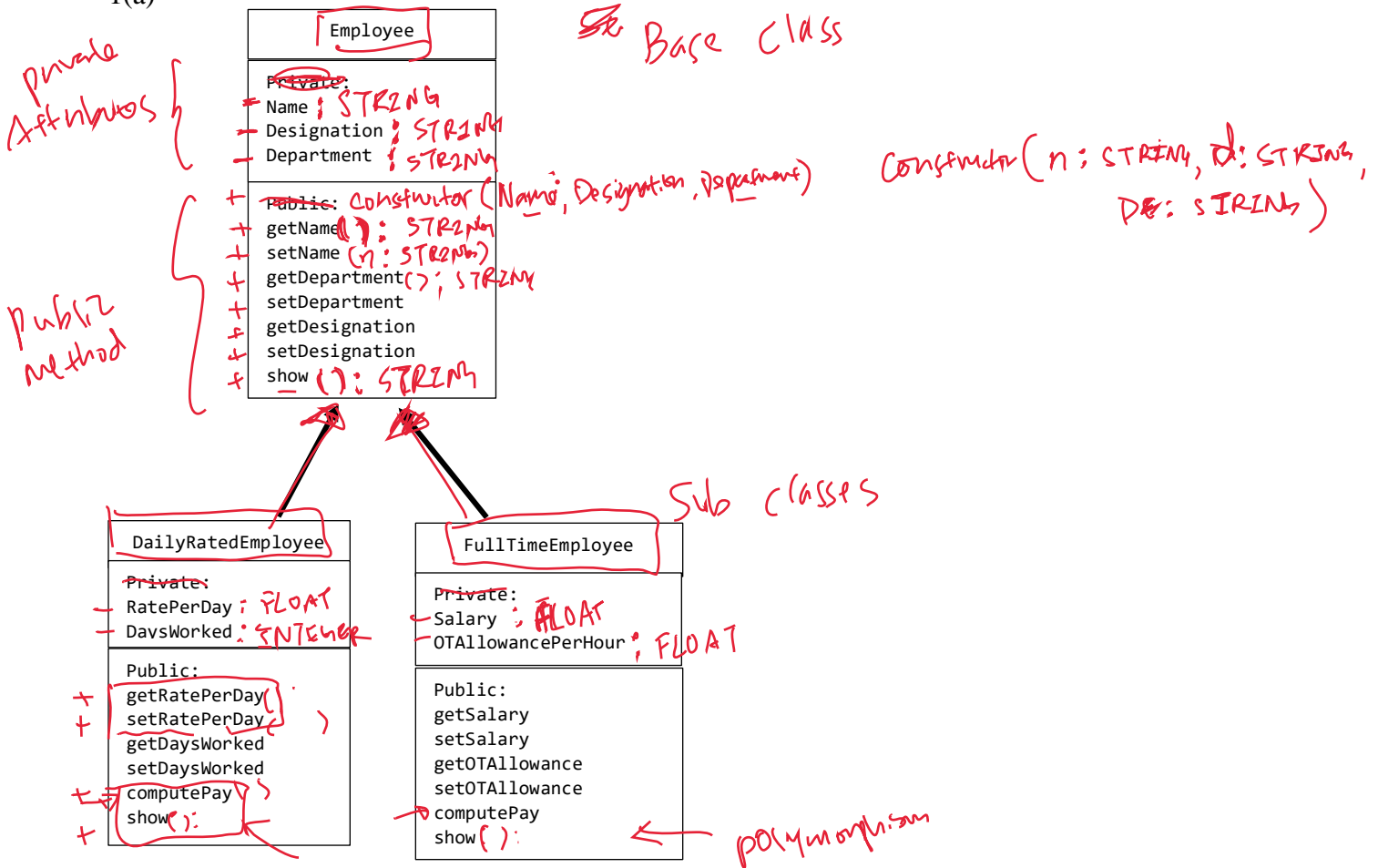
[2]

- (c) Explain the term **polymorphism** and how it is applied in your design in (a).

[2]

Solution Guide:

1(a)



[4M] 3 classes – 1 superclass, 2 subclasses with private attributes and public get/set methods for the attributes. Arrow from subclass to superclass to show

[1M] – Inheritance - Arrow from subclass to superclass to show inheritance

[1M] – Polymorphism – show method for all 3 classes

(b) The sub-classes (DailyRatedEmployee and FullTimeEmployee) inherit all the attributes and methods of the Employee class. The methods **getName, setName, getDepartment, setDepartment, getDesignation, setDesignation** are **inherited without changes to the implementation and the no coding are required**, hence code reused is achieved. [2M]

(c) Polymorphism refers to an object's ability to take different forms. It allows subclasses to have methods with the same name as methods in their superclasses. It gives the ability for a program to call the correct method depending on the type of object that is used to call it. [1M]

invoke
 E1.show()
 D1.show() → Name
 Design
 DailyRate
 No of days

additional or method's explain

E1 = Employee (Name, ...)
 D1 = Daily Rated Employee (n, ...)

Elaborate (address e ques)

technical term.
overrides
 overwrite

Explain how this is shown in your design

The method *show* in both the subclasses **overrides** the superclass *show*. If the subclass object (DailyRatedEmployee) is used to call *show*, then the subclass's version of the method is invoked. If the superclass object (Employee) is used to call *show*, then the superclass method will be invoked.

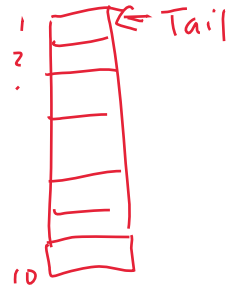
This shows polymorphism in action since the method [1M] '*show*' takes on different forms.

2. 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 operations.

Queue: 1-dimensional array with index 1 to 10 fixed size.

Head: pointing to the index of the first item in the queue

Tail: pointing to the index of the next item that is inserted



(a) Describe an algorithm, using pseudocode, to insert a new item NewItem into the queue. [4]

(b) Describe an algorithm, using pseudocode, to delete an item from the queue. [6]

(c) Peter intends to use the pseudocode $\text{Length} \leftarrow \text{Tail} - \text{Head}$ to find the length of the queue. Give an example to explain why he fails. Write down the correct pseudocode to find the length. [3]

(d) This data structure can also be implemented using linked list. Give **one** advantage and **one** disadvantage of linked list over array implementation. [2]

Solution Guide:

2(a)

```
01 IF Head = Tail
02   THEN
03     OUTPUT 'No more room to add item'
04   ELSE
05     Queue[Tail] <- NewItem
06     Tail <- Tail + 1
07     IF Tail = 11 (overflow)
08       THEN
```

```

09             Tail <- 1
10     IF Head = 0 (handle the empty case)
11         THEN
12             Head <- 1
13     ENDIF
14 ENDIF

```

[1] line 01 for correct condition of full queue

[1] line 05&06 for insertion

[1] line 07-10 for circular array

[1] head = 0 case

(b)

Method 1

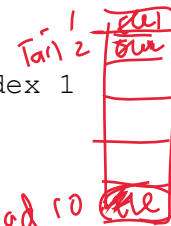
```

01 IF Head = 0
02     THEN
03         OUTPUT 'Empty queue'
04     ELSE
05         IF (Tail = Head + 1) OR (HEAD = 10 AND Tail = 1)
06             # queue with one element
07             THEN
08                 Head <- 0
09                 Tail <- 1
10         ELSEIF HEAD = 10 #new head circulates to index 1
11             THEN
12                 Head <- 1
13         ELSE
14             THEN
15                 Head <- Head + 1
16         ENDIF
17 ENDIF

```

} empty queue

ELIF



[1] line 01 for correct condition of empty queue

[2] line 05 for condition of one-element queue

[1] line 08&09 for reset Head and Tail

[1] line 10-12 for circular array

[1] line 15 for updating Head

Method 2

```

01 IF Head = 0
02     THEN
03         OUTPUT 'Empty queue'
04     ELSE
05         Head <- Head + 1

```

```

06      IF (Tail = Head) OR (HEAD = 11 AND Tail = 1)
07      # queue with one element
08      THEN
09          Head <- 0
10          Tail <- 1
11      ELSEIF HEAD = 11 #new head circulates to index 1
12      THEN
13          Head <- 1
14      ENDIF
15 ENDIF

```

[1] line 01 for correct condition of empty queue

[1] line 05 for updating Head

[2] line 06 for condition of one-element queue

[1] line 09&10 for reset Head and Tail

[1] line 11-13 for circular array

(c)

For example, in this case, the queue contains items in index 7, 8, 9, 10, 1, 2, so the length is 6, but Tail
 - Head = -4

Index	
1	
...	
3	Tail
...	
7	Head
...	
10	

Correct Statement:

```

IF Head = 0
  THEN
    Length <- 0

  ELSEIF Head < Tail
    THEN
      Length <- Tail - Head
  ELSE
    THEN
      Length <- 10 + Tail - Head

```

ENDIF

Each case with correct condition + calculation gives 1 mark. If students only get 2 cases correct, still give 2 marks. But tutors must explain all cases when returning the paper.

(d)

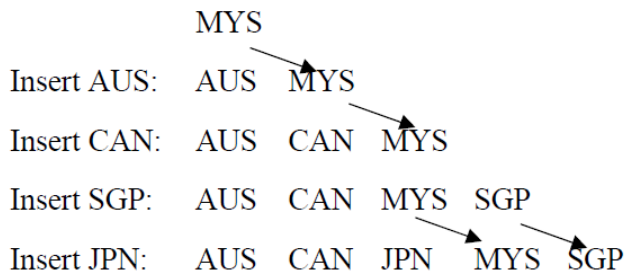
Advantage: linked list does not need to handle the circular case. Linked list provides dynamic spaces and provides bigger size for queue

Disadvantage: linked list requires more memory space to store the pointer for every item



Q3

(a) MYS, AUS, CAN, SGP, JPN.



General sense:

CL \Rightarrow Access element via traversing through LL

Array \Rightarrow Access element via index no.

(b) (i) A: UpperBound - 1 B: List [Posn + 1]

C: ~~False~~

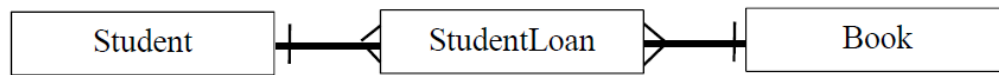
D: Temp

(ii) 2 passes. The list is mostly sorted in ascending order. Once there is no swap in ^{the 2nd} pass, the sorting process is done.

Question 4

(a) The file contains repeated information. For example: student name, phone number, title are repeated and stored in multiple records.

(b)



(c)

Book (BookID, BookTitle)

Student (StudentID, StudentName, PhoneNumber)

StudentLoan (LoanID, StudentID, BookID, DateBorrowed, DateDue, DateReturned)

OR StudentLoan(StudentID, Book ID, DateBorrowed, DateDue, DateReturned)

(d)

(i) **SELECT** StudentName, PhoneNumber **FROM** Student

(ii) **SELECT** Book.BookTitle, StudentLoan.DateBorrowed, StudentLoan.DateReturned

FROM StudentLoan **INNER JOIN** Book **ON** StudentLoan.BookID= Book.BookID

WHERE StudentLoan.StudentID = 3 ;

OR **SELECT** Book.BookTitle, StudentLoan.DateBorrowed, StudentLoan.DateReturned
FROM StudentLoan, Book **WHERE** StudentLoan.bookID= Book.BookID
AND StudentLoan.StudentID = 3

(e) any two from below + **student data** related to the question (e.g. phone number)

1. Consent Obligation - Only collect, use or disclose personal data when an individual has given his/her consent.

2. Purpose Limitation Obligation - An organisation may collect, use or disclose personal data about an individual for the purposes that a reasonable person would consider appropriate in the circumstances and for which the individual has given consent.

3. Notification Obligation - Notify individuals of the purposes for which your organisation is intending to collect, use or disclose their personal data on or before such collection, use or disclosure of personal data.

4. Access and Correction Obligation - Upon request, the personal data of an individual and information about the ways in which his or her personal data may have been used or disclosed in the past year should be provided. Organisations are also required to correct any error or omission in an individual's personal data upon his or her request.

5. Accuracy Obligation - Make reasonable effort to ensure that personal data collected by or on behalf of your organisation is accurate and complete, if it is likely to be used to make a decision that affects the individual, or if it is likely to be disclosed to another organisation.

6. Protection Obligation - Make security arrangements to protect the personal data that your organisation possesses or controls to prevent unauthorised access, collection, use, disclosure or similar risks.

7. Retention Limitation Obligation - Cease retention of personal data or remove the means by which the personal data can be associated with particular individuals when it is no longer necessary for any business or legal purpose.

8. Transfer Limitation Obligation - Transfer personal data to another country only according to the requirements prescribed under the regulations, to ensure that the standard of protection provided to the personal data so transferred will be comparable to the protection under the PDPA.

9. Accountability Obligation - Make information about your data protection policies, practices and complaints process available on request.

(f)

Backup	Archive
Enables rapid recovery of live, changing data	Stores unchanging data no longer in use but must be retained
Multiple copies of data	One copy
Restore speed: crucial	Retrieval speed: not crucial
Short Term Retention Retained for as long as data is in active use	Long Term Retention Retained for required period or indefinitely
Duplicate copies are periodically overwritten	Data cannot be altered or deleted

(g)

[1] Version Control

1. Ease collaboration across distributed teams -a version control system makes it much easier to share code changes and coordinate the work of the various team members who are responsible for the database. The ability to rapidly share and manage changes makes it particularly important for teams based in different locations, and evidence shows that teams are increasingly distributed.

2. Gain better visibility of the development pipeline -a version control system provides an overview of what development work is going on, its progress, who's doing it, and why. It also maintains detailed change histories and can be associated with issue tracking systems.

3. Ability to roll back or retrieve previous versions- it provides an efficient mechanism for backing up the code. Because the history it provides is incremental, version control lets developers explore different solutions and roll back safely in the case of errors.

4. More readily demonstrate compliance and auditing -The change tracking provided by version control is the first step to getting ready for compliance, and an essential step in maintaining a robust audit trail and managing risk. Compliance auditors will require an organization to account for all changes and detail all those with access to it.

Naming Convention

1. Readability is important. Readability helps in figuring out what the code does in less time.
2. A common naming convention that everyone agrees to follow will result in developers, reviewers and project managers communicate effectively with respect to what the code does.
3. Naming conventions result in improvements of communication, code integration, consistency and clarity and lead to predictability and discoverability (easier to find files)

5(a)

		Rules							
Conditions	First timer	Y	Y	Y	Y	N	N	N	N
	Spent at least \$1000 in past 3 months	Y	Y	N	N	Y	Y	N	N
	Inactive for 1 month	Y	N	Y	N	Y	N	Y	N
Actions	\$5 discount	X	X	X	X			X	
	10% discount	X	X			X	X		
	Additional 5% discount	X				X			
	No discount								X

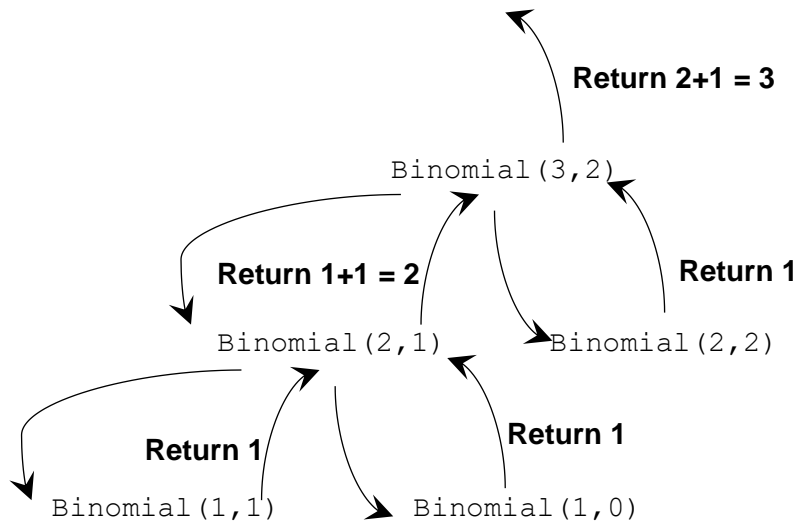
1(b)

		Rules				
Conditions	First timer	Y	N	N	N	N
	Spent \$1000	N	Y	Y	N	N
	Inactive	N	Y	N	Y	N
Actions	\$5 discount	X			X	
	10% discount		X	X		
	Extra 5%		X			
	No discount					X

6(a)

A recursive function is one that calls itself. Line 06 makes it recursive.

6(b)



(c) Any of the following:

- $R > N$

- $R < 0$

- $N < 0$

Q7[14]

(a) LAN – small geographical area, Private any 1

[2] WAN – large geographical area, Public any 1

(b) End-devices, use's laptops, mobile devices connect to LAN [1]

[2] LAN connect to a ISP network using a WAN link provided by the service provider[1]

(c) A set of rules that determine exchange of data between 2 nodes in the network [1]

[4] TCP/IP consists of 4 layers of protocols [1]

Application layers deals with end application communicating with end application

Transport layer determines whether the communication is reliable or best-effort delivery TCP/IP UDP

Network layers determines how packets are routed in the network: For

Physical layers determine the physical hardware involves in the communication

Any 2 of the layer correctly described [2]

- (d) Server connected to a LAN [1]
- [4] LAN connects to a ISP/Internet[1]
- Student PC connects to the Internet [1]
- Teacher PC connects to the Internet [1]
- (f) DNS name consists of a hostname and a domain name, it is unique in the public
- [2] Internet [1]
- Obtain/register a valid domain name from a service provider [1]

Q8 (b)(i) Data that can (uniquely) identify a living person;

(b)(ii)

Linked to context: (MAX 2)

Data could be used to track location (and activities) of a person; Data links a person to a specific location and car at a (specific) time; Number plates might not be recognised accurately (suggesting, incorrectly, a car was at a particular location);

General points: (MAX 1)

Q9

2

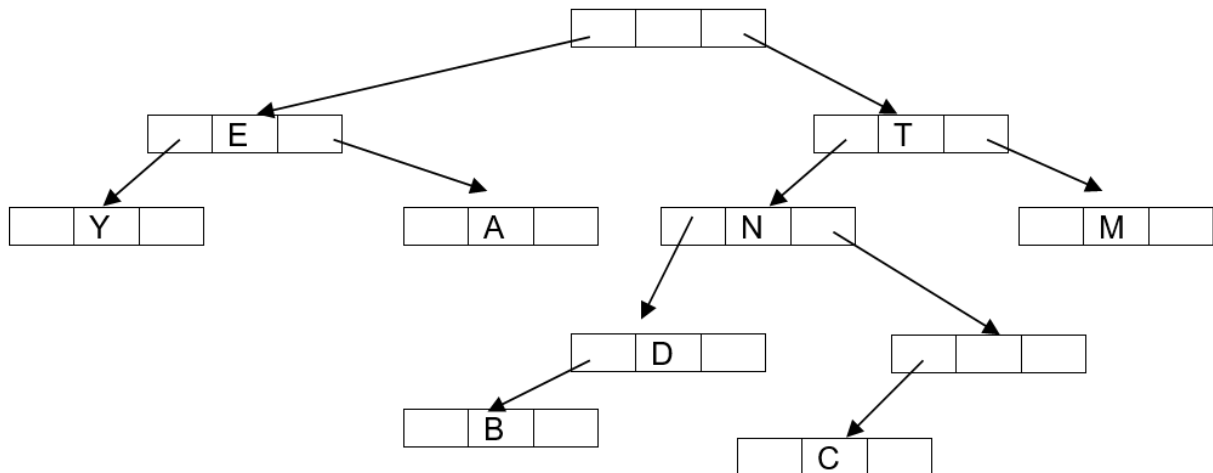
(a) What are the Morse codes for the letters N and Y?

[2]

Y = . . (dot dot)

N = - . (dash dot) (1 mark each)

(b) Draw a diagram of the binary tree which shows clearly the position of the letters D, C and B in the tree. [3]



- (c) Explain why this binary tree representation is not the most suitable data structure for performing English to Morse code conversion. Describe a better alternative, and explain how the Morse code of a letter could be found. **[5]**

Why not the most suitable ? (1 mark)

require a tedious tree traversal algorithm to find the morse code equivalent of a letter. Some kind of total tree traversal method would be needed.

Alternative data structures: (3 marks for any one structure)

(i) Linked list: not recommended as it does not improve efficiency at all.

(ii) 2 one-dimensional arrays: one containing the letters while the other containing the Morse code for each letter.

- linear search of “letter” array to locate the index from which the equivalent morse codes can be found.

- binary search instead . (better efficiency)

(iii) store the Morse codes in a 1D array indexed by the letters of the array if possible, and by codes A = 1, B = 2, C = 3, ...Z= 26. The morse code can be retrieved directly from its index value (or from its index value after calculation).

A	. -
B	- . . .
C	- . - .
D	- . .

Qu						Marks	
02	1	All marks AO2 (apply)					5
		Count	HexString	Number	HexDigit	Value	Output
		1	"A2"	0	"A"	10	
				10	"2"	2	
				162			162
		2	"1G"	0	"1"	1	
				1	"G"	-1	
				15			15
		Mark as follows:					
		1. Count running over the values 1, 2 with correct sequence of values for HexString ("A2", "1G");					
		2. The correct sequence of values in Number column (0, 10, 162, 0, 1, 15);					
		3. The correct sequence of values in HexDigit column ("A", "2", "1", "G");					
		4. The correct sequence of values in Value column (10, 2, 1, -1);					
		5. The correct sequence of values in Output column (162, 15);					
		A. repeating values in first two columns					
		A. "1G" before "A2"					
		A. string values without quotes					
02	2	All marks for AO2 (analyse)					2
		1. invalid character produces value -1 from subroutine;					
		2. -1 should not be used to calculate // deal with -1 seperately // using -1 gives a misleading result;					
		3. final output should be -1 / error message;					
		MAX 2					