# **C Z** I S **E Z**

GOOD LUCK BUDDY

# Do What Singapore Preaches

⇒ Memorize & Regurgitate

take everything w a pinch of salt tho it’s mostly student submitted - teacher approved answers so not like 100% correct but ya

and also turn a blind eye on our ugly ass drawings of diagrams n shit lmao

also edit access is switched on ~~so have fun~~

# Part 1: Algorithms and Design

## Chapter 1: Fundamental Algorithms

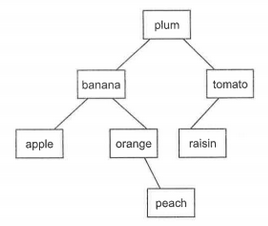
## Chapter 2: Data Representation and Structures

## Chapter 3: Modularity

## Chapter 4: Programming

**2014**

**2.** Consider the following binary tree:



**(a)** List the nodes, in order, that are visited for a post-order traversal. [2]

|  |
| --- |
| **Left - Right - Root:** # Check if Root last, else something is wrong  Apple - Peach - Orange - Banana - Raisin - Tomato - Plum |

pre-order (Root, Left, Right) - plum - banana - apple - orange - peach - tomato - raisin

**(b)** List the nodes, in order, that are visited for an inorder traversal. [2]

|  |
| --- |
| **Left - Root - Right:**  Apple - Banana - Orange - Peach - Plum - Raisin - Tomato |

**(c)** What property is exhibited by the list of items produced in **part (b)**? [1]

|  |
| --- |
| It is sorted by alphabetical order. |

**(d)** Describe an algorithm, using pseudocode, to perform a binary tree search. The output   
 should state whether or not the item is present in the tree. [5]

|  |
| --- |
| *4# It’s pseudocode, not writing an essay. - Xu Hua*  BinarySearch(root, data)  IF data equals(=) root data  Output present  ELSE IF (not root left subtree) and (not root right subtree)  Output not present  ELSE IF data > root data  IF root’s right subtree exists  search root’s right subtree  ELSE  Output not present  ELSE  IF root’s left subtree exists  search root’s left subtree  ElSE  Output not present  ENDIF |

**2014**

**5.** A software developer is given the task of producing software for a college. The software   
 will help to manage information about what students do after finishing at the college.

The destination of each student after college is classified in one of three possible ways:

* University
* Employment
* Other

The college wishes to store: profiles for their CIA database

* name
* number of A Level passes
* destination (U / E / O)
* university attended
* main subject studied at university
* type of employment
* what students do when their destination is classified as 'O'

The software developer will use an object oriented approach to developing a solution.

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

* suitable classes with appropriate properties and methods
* inheritance
* polymorphism

[6]

|  |
| --- |
|  |

**(b)** Explain how your solution to (a) demonstrates software reuse. [2]

|  |
| --- |
| **Definition of inheritance:**  Inheritance allows common methods and data to be in the parent’s class, so that the common methods and methods do not need to be rewritten in the child classes. For example all students will have a name data and also a corresponding method to acquire the data.  **Definition of polymorphism:**  Invoke the classes using the same method name. Polymorphism allows common methods with different functions to be called even though they are different type of objects, this makes the objects abstract so common calling methods can be used to call different objects and there is no need to write specific code to handle each child. For example, we can use the display() method for students that are still going to university and also employment while getting different output. |

The data on the students is to be stored in a serial text file called STUDENT.DAT. Each line of the file has the same structure:

<Name><NoOfPasses><Destination><University><MainSubject><EmpType><Other>

with the string NULL stored where appropriate.

**(c)** Write an algorithm, in pseudocode, to read data from STUDENT.DAT and to output the

following:

* total number of students going to university
* average number of passes for the students going to university
* total number of students

|  |
| --- |
| Open STUDENT.DAT in read mode  # Declare variables  NumberOfStudentsGoingUni ← 0  NumberOfPasses ← 0  NumberOfStudents ← 0  FOR each line in the file  IF university <> NULL  NumberOfStudentsGoingUni ← NumberOfStudentsGoingUni + 1  IF NoOfPasses <> NULL  NumberOfPasses ← NumberOfPasses + NoOfPasses  ENDIF  ENDIF  NumberOfStudents ← NumberOfStudents + 1  ENDFOR  Output NumberOfStudentsGoingUni  Output NumberOfPasses divided by NumberOfStudentsGoingUni  Output NumberOfStudents |

[7]

**2014**

**6.** A function is to be written that returns the sum of all values held in an array that are greater than a minimum value. The function will be used with arrays of varying size, but never more than a maximum of 50 000 elements.

A first attempt at writing the program code for the function is given below:

l. FUNCTION TotalSum(Results : ARRAY[50000] OF REAL,

ArraySize : INTEGER, MinValue : REAL)   
 RETURNS REAL

2. DECLARE Sum, Counter : INTEGER

3. DECLARE Temp : Real

4. Sum = 0.0

5. FOR Counter = 1 TOO ArraySize

6. Temp = Results[Counter]

7. IF Temp > MinValue THEN Sum = Sum \* Temp

8. ENDFOR

9. RETURN Sum

10. ENDFUNCTION

The function is included in a program specifically written to test the function. The main   
 program outputs the value returned by the function. A compiler was used to compile the   
 source program.

**(a)** The compiler reported an error at line 5 in the function. Identify the error and explain   
 why it was flagged as a syntax error. [2]

|  |
| --- |
| * Keyword TO as part of the FOR loop construct was misspelled as TOO * flagged as syntax error as this violates the grammatical rule of the programming language by not having the correct reserved word in place   Note: Corrected statement should be FOR Counter = 1 TO ArraySize |

**(b)** The compiler also reported an error at line 4. State the type of error reported by the   
 compiler justifying your answer. [2]

|  |
| --- |
| * Semantic error: sum is declared to be an integer data type yet a float value 0.0 is assigned to it * Semantic error is an error arising from the wrong meaning attached to a program statement and is flagged by the language translator   Note: SUM should be declared as REAL data type |

The errors indicated in **parts (a)** and **(b)** were corrected. A successful compilation   
 produces executable code. When the code was executed, the program failed to   
 complete and reports an error at line 7.

**(c) (i)** State the type of error that occurred. Justify your answer. [2]

|  |
| --- |
| * Arithmetic runtime error (overflow error) * Caused by the result of the arithmetic operation being too large to fit into a finite number of bits of the appropriate data type, leading to a wrong (potentially negative) answer stored.   # e.g overflow, underflow (typically will not generate an error), division by 0 (considered as arithmetic errors and will signal the program to stop but underflow error may not generate a runtime error and usually will store 0 as the estimated value). |

**(ii)** The error described in **part (c) (i)** depends on the detection of another type of error.   
 Name this other type of error. How should the code be changed to correct this   
 error? [2]

|  |
| --- |
| * Logic error which signify a wrong representation of intention which is not flagged by language translator but often will produce the wrong result: computation of Sum is intended to be an addition, instead of a multiplication * Correct code should be Sum = Sum + Temp (follow question case convention!)   **~~Fix:~~** ~~SUM = SUM + TEMP~~ |

When the program finally runs without error, the test plan needs to be completed. The test   
 plan uses data that tests different sizes of array, different array values and different   
 minimum values.

The array TempArray is used in the main program as the array to be processed.

**(d)** Each element of TempArray stores a random value between 1.0 and 10.0.

**(i)** Explain why the function call:

TotalSum(TempArray, 1000, 5.0)

is not an appropriate black box test. [2]

|  |
| --- |
| * Black box testing evaluates if a program's actual output matches its expected output given a well defined specified set of inputs, without explicit knowledge of the program statements * As the array values are randomly generated, it is not possible to provide the expected output for comparison with the actual output generated by the function call * For a valid black box test, the input should be fixed values adhering to the program specification and not randomly generated   ~~Previous student-proposed answer:~~   * ~~As the program is designed to work for an array sizes up to 50000, black box testing should be built around the specifications of the program and hence an array size of 1000 would not sufficiently test the full specifications of the program. It also does not test for erroneous data.~~ * ~~While the test may on its own is a valid black box test, it is insufficient on its own to fully test the range of the program, and hence it should be part of a set of test cases to fully check the function of the program.~~ |

**(ii)** Explain why the function call:

TotalSum(TempArray, 10, 10.5)

is not an appropriate white box test. [2]

|  |
| --- |
| * White box testing is a structural test by the programmer with knowledge of the program statements and aims to evaluate all possible control paths of a program * As 10.5 is not within any of the array values between 1.0 and 10.0, the conditional control path IF Temp > MinValue THEN Sum = Sum \* Temp will never be reached, hence the function will not constitute an appropriate white box test   ~~Previous student-proposed answer:~~  ~~White box testing means all branches are tested.~~  ~~This function call only test one specialised path.~~ |

**(e)** lf each element of TempArray stores the value 1.0, state a function call that will be an   
 appropriate black box test. Justify your answer. [3]

|  |
| --- |
| * TotalSum(TempArray, 50000, 1.0) will be an appropriate black box test as * This evaluates the actual output of the program against the expected output of 0, given the valid extreme/boundary values of ArraySize and MinValue * The maximum extreme value of 50000 for ArraySize will also test the program for linear runtime efficiency   Student-proposed answer:  To test extreme data values: TotalSum(TempArray, 50000, 0)  This black box test will test the edge cases of the program, which specifies that it can run up to a maximum array size of 50000. The usage of 0 as the minimum value allows us to calculate a result manually (i.e. 50000 \* 1.0 = 50000) that can be easily verified with the program’s output . If the results are the same, then the program passes the black box test. |

**2013**

**3** A hash table has an index range of 1 to 900. The following pseudocode describes an   
 algorithm for searching the table using the hashing function Hash. It is assumed that the   
 key is present in the table.

1. Index <- Hash(Key)

2. WHILE Table[Index, 1] <> Key

3. Index <- Index + 1

4. ENDWHILE

5. Value <- Table[Index, 2]

**(a)** Explain the purpose of:

**(i)** line 3

|  |
| --- |
| * **Collision** has occurred, **linear probing** (increment by 1) is used to resolve collision; advance to the next hash table address slot and check if the record key matches target key. |
| # Ambrose (Senior’s answer - lacking keywords)   * When the data stored in that particular index of the table obtained by the hash key does not correspond with the data of the variable Key, the index increments by 1 to check the next possible record if it is the record containing the key. This therefore allows linear probing to be carried out. |

**(ii)** line 5

|  |
| --- |
| * Matching record key is found and the second column containing the desired record value is retrieved, assigned to variable Value (for use in subsequent program statements or returned to calling program) |

[4]

**(b)** Describe a problem that might occur with a key which, when hashed, produces an   
 index of 900. [2]

|  |
| --- |
| * If there is a collision at this position, it will enter the while loop (line 2) and **undergo linear probing/increment index by 1** (line 3). This will assign Index to the value 901, which is **out of range**. Program will fail with an out of range (runtime) error. |

**(c)** What modification to the algorithm is required to overcome this problem? [3]

|  |
| --- |
| * Implement a **wrap around technique** for the linear probing (increment by one) method, such that any increment of the index exceeding 900 will start at the front of the array again. * ~~i.e. change line 3 from Index <- Index + 1 to Index <- (Index + 1) mod 901 // assume 1-based array~~ * ~~where mod is the modulo arithmetic (remainder) operator~~ * i.e. Add the following conditional statement after line 3: * if Index > 900 * Index <- 1 |
| Alternatively,   * Within the while loop, implement an if-else statement to account for the extreme index value of 900. * If index is 900 (extreme index value), index is reset to 1/to start of range (instead of linear probing). Else, (normal index values) undergo linear probing. |

**(d)** Explain how a new item can be added to this hash table. [4]

|  |
| --- |
| // assume 1-based array  Index <- Hash(NewKey)  WHILE Table[Index, 1] <> -1  Index <- Index + 1  IF Index > 900  Index <- 1  ENDWHILE  Table[Index, 1] = NewKey  Table[Index, 2] = NewValue   * use Hash function to determine hashed address and * check if it is occupied (assume it contains -1 if not occupied) * while occupied, increment by 1 to find next available address slot * set column 1 to new key and column 2 to new value * if no more empty slots, return error |

**2013**

**6** The ASCII code for the character 'Z', expressed as a denary integer, is 90.

**(a)** Express the denary integer 90 as:

**(i)** an eight-bit binary number

|  |
| --- |
| * 9010 = 10110102 * Perform repeated division by 2 (binary base) and concatenate remainders (0 or 1) in reverse. |

**(ii)** a hexadecimal number

[2]

|  |
| --- |
| * 9010 = 5A16 * Perform repeated division by 16 (hexadecimal base) and concatenate remainders (0 to F) in reverse. * 90 mod 16 = 10 (i.e. A) r 5 * Thus join 5 and A |

**(b)** Give two reasons why hexadecimal numbers are used in computing. [2]

|  |
| --- |
| * Each hexadecimal digit is equivalent to 4 binary digits, hence conversion is **efficient** and more **human-friendly** (takes up less space visually and easier to read) than raw binary 1s and 0s. * Hexadecimal digit is base 16 and is in powers of 2, hence it is an exponential of binary which is **easily manipulated.** * **Easily processed by computers which operate using binary** **(on/off) states.** |

**(c)** State the ASCII code for 'X' in denary. Explain your answer. [2]

|  |
| --- |
| * 88, as 'X' is 2 lexical positions from 'Z' which is 90.  (and can be derived by subtracting 2 from ASCII code for 'Z'.) |

**(d)** Explain why the Unicode encoding system has replaced ASCII. [2]

|  |
| --- |
| * ASCII is **7-bit encoding** of characters (+1 bit for parity) based on the English alphabet and can only represent 27/128 characters, limited to only digits, uppercase and lowercase english alphabet characters, punctuation symbols and some control characters. * As information exchange over the Internet and networks have become **global**, it is necessary to represent many more **international languages beyond English**, the only language ASCII contains. Furthermore, **some of the ASCII control characters are now obsolete**. * A **multi-byte encoding scheme** such as Unicode has hence become the universal standard, as it is able to represent **all possible characters from all writing systems** in the world. |

**(e)** Describe a method of storing strings of characters of variable length in a computer. (definition + pros/cons + evaluation + examples in the context)

[2]

|  |
| --- |
| * **Comma separated variables** making use of a suitable delimiter character can be used to store strings of characters of variable length * e.g. Lim Ah Seng,123 Sesame Street * Joy Tan,10 Rhu Ave * This **optimises storage** compared to fixed length record fields where **maximum space needs to be allocated** for the longest string. * Most computing system have built-in facilities to read and extract the individual fields separated by the delimiter character   # Addendum:   * Some relational database system uses VARCHAR (Variable Characters) to represent variable length strings eg VARCHAR(255) can store characters of variable length up to well defined upper limit of 255 characters. |

**2012P1**

**3** An array, A, of integers stores the following values:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A[0] | A[1] | A[2] | A[3] | A[4] |
| 9 | 4 | 2 | 7 | 6 |

The following algorithm sorts the contents of the array A into ascending order.

i = 4

REPEAT

FOR j = 0 to i - 1 DO

IF A[j] > A[j + 1] THEN swap A[j] and A[j + 1]

ENDFOR

i = i - 1

UNTIL i = 1

**(a)** Copy and then complete the following trace table to show how the algorithm sorts the   
 contents of array A into ascending order.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| i | j | A[0] | A[1] | A[2] | A[3] | A[4] |
|  |  | 9 | 4 | 2 | 7 | 6 |
| 4 | 0 | 4 | 9 | 2 | 7 | 6 |
| 4 | 1 | 4 | 2 | 9 | 7 | 6 |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

[4]

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | i | j | A[0] | A[1] | A[2] | A[3] | A[4] | |  |  | 9 | 4 | 2 | 7 | 6 | | 4 | 0 | 4 | 9 | 2 | 7 | 6 | | 4 | 1 | 4 | 2 | 9 | 7 | 6 | | 4 | 2 | 4 | 2 | 7 | 9 | 6 | | 4 | 3 | 4 | 2 | 7 | 6 | 9 | | 3 | 0 | 2 | 4 | 7 | 6 | 9 | | 3 | 1 | 2 | 4 | 7 | 6 | 9 | | 3 | 2 | 2 | 4 | 6 | 7 | 9 | | 2 | 0 | 2 | 4 | 6 | 7 | 9 | | 2 | 1 | 2 | 4 | 6 | 7 | 9 | | 1 | 0 | 2 | 4 | 6 | 7 | 9 | |

**(b)** Name the sort method being used. [1]

|  |
| --- |
| * Bubble sort (largest number bubble/float to the top/right at the end of each pass) |

**(c)** Explain, using the trace table, why this sort is inefficient. [8]

|  |
| --- |
| * In the trace table, it can be observed that even though the elements in the array A have already been sorted, from rows 5-6 and rows 8-11, yet the algorithm still continues to make ~~swaps, rendering it as inefficient~~ subsequent comparisons which constitute **unnecessary work done** * This is because the algorithm uses two loops (repeat and for) with **fixed/static bounds with no conditional checks** for the sortedness of the array. * Hence the loop body is performed from initial to terminating conditions, **regardless of whether sorting has already been achieved**. |

**(d)** Amend the algorithm so that that sort process can overcome the inefficiency identified   
 in **(c)**. [4]

|  |
| --- |
| * ~~use a boolean flag to check if there is any swap in each pass~~ * ~~if there is no swap, array is already sorted and hence algorithm can terminate~~ * Introduce a **boolean flag** ‘swapped’ first initialised to False (assume array is sorted). * Inside the inner FOR loop, set swapped to True if a swap takes place. Since there is a swap, this implies array is not sorted during that pass. * if within a pass of the for loop, there is no swap, indicates that array has been sorted and outer repeat loop **can terminate** * amended algorithm:   i = 4  REPEAT  swapped = False // assume sorted  FOR j = 0 to i - 1 DO  IF A[j] > A[j + 1] THEN  swap A[j] and A[j + 1]  swapped = True  ENDIF  ENDFOR  i = i - 1  UNTIL not swapped // will terminate once array sorted |

**(e)** Give **two** different sets of test data that could be used to test the amended algorithm   
 in **(d)**.  
 Explain the purpose of each set of test data. [4]

|  |
| --- |
| * already sorted data eg 1, 3, 5, 7, 9 - check that algorithm **terminates in first pass** since no swapping takes place. * partially sorted data with most numbers in place eg 5, 3, 7, 9, 1 - check that algorithm **terminates early** when numbers are already sorted without undergoing unnecessary further passes. |

**SP9597**

**2 (a)** Explain the difference between data validation and data verification. [2]

|  |
| --- |
| * Validation is the **automated checking of data** to ensure data values are **acceptable / reasonable** * ie. Type check for numeric data, range check fixed length data * ~~Verification is the process of ensuring data transfer is successful by performing double entry matching / screen matching~~ * Better: verification is process of ensuring **transferred data matches source data** (can be manual (double entry matching / screen matching) or automated (checksum / control totals / exact match check))   to clarify:  Double entry matching means inputting data a second time to verify that it matches the data entered during the first time. maybe to avoid ambiguity, write verbosely rather than use this term.  Two pass verification aka double data entry is one specific example of manual data verification. i would suggest that u can rename double entry matching to double data entry if it makes the distinction clearer for you. the point is: verification in present day context encompasses both manual and automated forms.  checksum / control totals can be used for both validation (which is always automated) and automated verification. (check digit is used strictly for validation). an example of checksum / control totals being used for automated verification is to generate a checksum for a downloaded file/program and confirm that it matches the one provided on the server (eg hover over the checksum md5 and sha1 columns at<https://www.apachefriends.org/download.html>)  it will be instructive to provide the example in context to avoid confusion. because in the case of file/download example, the matching of checksum aligns with the definition of verification - matching transferred data (client generated checksum) with source data (server provided checksum).  For exact match check, an example of verification is confirming that the user supplied password matches the system stored password (both should be encrypted). always align with definition i.e. destination matches source.  Do study the context. if it is manual verification, then checksum/control totals/exact match should not be used. in the question, since the question comes before a context, it is not wrong to give a general purpose definition. |

A car insurance company reads allows customers to insure their car using one of the two   
 methods:

* method A: by using the Internet or
* method B: by using the telephone to talk to a sales representative

(b) For method A describe how the car registration could be validated. [2]

|  |
| --- |
| * registration number has particular length / format / existence check / check digit * these are checked by **client-side and server-side** program code before and after data is submitted to web server to ensure data is valid   # data may be corrupted during the transmission, therefore it is necessary to check the data by client-side and server-side program code  # Client side validation saves bandwidth through use of javascript to check for non-matching data (length, alphabets, numbers not matching etc.)  # Server side validation is required to check for the integrity of the data, to ensure that the data is not altered or lost accidentally or intentionally during the transfer (e.g. corruption due to noise, hackers modifying the data) |

(c) For method B describe how the car registration could be verified. [2]

|  |
| --- |
| * Salesperson asks customer to **repeat** the registration number / salesperson and reads it back * Salesperson checks that ~~two registration numbers are the same / customer confirms correct number~~ the provided registration number has a matching number in the source database. |

(d) The sales representative enters information provided by new customers into the   
 computer system using a graphical user interface. Some of the information required   
 includes:

* customer salutation
* customer name and address
* customer gender
* car registration number

For this application design a simple screen layout which makes use of appropriate   
 graphical user interface controls. [3]

|  |
| --- |
| * Please draw diagram with appropriate placement (Should be rectangular box, should have a heading) and labelled controls * Suitable screen layout including submit/next or similar command button   + text boxes for customer ID, name, address and car registration   + customer salutation - drop down list with suitable items e.g. Mr, Mrs, Ms   + gender - radio buttons |

**SP9597**

**3** A recursive procedure for finding a value, SearchItem, in an ordered array X, is as   
 follows:

01 PROCEDURE Search(Low, High)

02 Middle <- (Low + High) DIV 2

03 IF X[Middle] = SearchItem THEN OUTPUT "found"

04 IF X[Middle] > SearchItem

05 THEN

06 Search(Low, Middle - 1)

07 ELSE

08 Search(Middle + 1, High)

09 ENDIF

10 ENDPROCEDURE

Note: the DIV operation returns an integer quotient after division e.g. 7 DIV 2 = 3

**(a)** Using the above procedure explain what is meant by a recursive algorithm. [2]

|  |
| --- |
| * **subprogram** that **calls itself** / subprogram that can repeat itself indefinitely / subprogram whose definition includes reference to itself (gisc note: not a precise definition) * subprogram that calls itself with a **smaller and similar structured** problem until terminating condition is met, subprogram then **backtracks** to original problem, combining solution along the way until solution to original problem is reached * The Search procedure will undergo recursion and call itself if the condition in line 4 is satisfied (seen in line 6 and 8) and will continue so until terminating case (condition in line 3 satisfied, or both conditions in line 3 and 4 are not satisfied) is reached. |

Array X has 15 elements and the subscripts start at 1.

**(b)** If the procedure was used to search the array X for the value stored at X[3] state the   
 calls to Search as the recursion executes. [4]

|  |
| --- |
| * Search(1, 15) * Search(1, 7) * Search(1, 3) * Search(3, 3) |

**(c) (i)** A program is to be written using this procedure. Explain why the search for an   
 item not present in the array would eventually result in the program crashing. [2]

|  |
| --- |
| * There is no code to **handle** item not present, recursion **continues indefinitely** * Each recursive call results in **use of storage** for current state of program * Eventually memory will be **exhausted** / heap exhausted / stack overflow   # 1 mark per point to a max of 2 |

**(ii)** Explain what changes need to be made to this procedure so that it handles the   
 case where SearchItem is not present in array X. [3]

|  |
| --- |
| IF Low > High THEN OUTPUT "item not found" : ELSE   * **Comparison** of Low and High * Output of suitable message * Exit / else * Code needs to be placed **at start of procedure**   # 1 mark per point to a max of 3 |

**SP9597**

**5** A programmer is going to write an application, using an object-oriented programming  
 language, which will store details of the vehicles available for hire from a vehicle rental  
 company. The company hires out both cars and vans. The conditions of the hire state   
 that cars can only carry passengers and vans can only carry loads. Properties identified   
 include:

* Registration code
* Maximum number of passengers
* Maximum load weight
* Type of vehicle

**(a)** Draw a diagram that shows how the properties could be distributed amongst a   
 number of classes. Include in your diagram any inheritance between the classes.   
 Also indicate some of the methods that would be required. One method should   
 demonstrate polymorphism. [6]

|  |
| --- |
| * 1 mark for 3 classes * 1 mark for correct inheritance shown (upward pointing arrows) * 1 mark for correct distribution of attributes * 1 mark for correct use of public and private * 1 mark for identification of appropriate methods * 1 mark for method exhibiting polymorphism |

**(b)** The company decide that they wish to also rent out minibuses for the carrying of  
 passengers. They decide to offer 6, 12 or 14 seater minibuses. Explain how this   
 affects the classes, properties and methods given in **(a)**. [5]

|  |
| --- |
| either   * minibus is neither car nor van * so new class Minibus with parent class Vehicle * new class has property of MinibusType / NoOfSeats * restricted to limited set of value * new class has methods for setting and getting new property * possible values for VehicleType increased to include Minibus   or   * minibus and car both carry passengers * only difference is maximum number of passengers * so new class PassengerVehicle with parent class Vehicle * new class has property of MaxPassengers * has methods for setting and getting MaxPassengers and Show * possible values for VehicleType increased to include PassengerVehicle |

**(c)** Explain, using this example, why encapsulation is an example of good programming  
 practice. [3]

|  |
| --- |
| * data and methods to manipulate data **held in same structure** * implementation details **hidden** and data **only accessible via class interface** * e.g. setting VehicleType includes check for valid value - handling new values does not require alteration of method calls |

**2012P1**

**5** A new computerised system will have records that include a date field. Two possible   
 formats for the date field are:

* a format of DD/MM/YYYY for example 23/06/2012
* a format of DD/ name of month /YYYY for example 23/JUNE/2012

All other fields in the record are of a fixed length.

**(a)** Explain why the first format would result in fixed length records whereas the second   
 format would result in variable length records. [4]

|  |
| --- |
| A fixed length record file is one where a given attribute has the same length across all records.  The first format follows a **fixed structure**, with ‘date’, ‘month’ and ‘year’ fixed to 2 characters, 2 characters and 4 characters long respectively, **regardless of the actual value** of the record.  Fixed length records are **easier for computers to process and for humans to read,** at the expense of **wasted storage space** where an attribute’s value does not take up the full allocated space. |
| A variable length record file is one where each attribute of a record can have **variable/unrestricted lengths**, and are joined together by a **delimiter** character to form the record.  The second format follows this structure, with the ‘month’ attribute able to have values of different lengths; for instance the length of the attribute for ‘JANUARY’ would be 7, and for ‘FEBRUARY’, 8.  Such records take up less space by **minimising wasted storage space**, but require **more complex processing and are harder for humans to read**, but present computer systems has *built-in/standard library routine to read/process such delimited records*. |

**(b)** The second format could be adapted so that a fixed length record would be possible.

Describe **two** different ways in which this can be done. [3]

|  |
| --- |
| 1. Fix the length of the ‘month’ attribute at 9, i.e. the length of the longest month name (SEPTEMBER).  If the data to be stored is less than nine characters long, fill up the remaining gaps with a placeholder character (e.g. space). 2. Abbreviate the names of the months to their 3 letter short-forms (i.e. JAN for January, FEB for February etc.) such that every month name is of the same length. |

The system developer decides to use the second format and to have variable length   
 records.

**(c)** Describe **two** methods of storing variable length records in a file. [4]

|  |
| --- |
| 1. **comma separated** variables making use of a suitable delimiter character can be used to store strings of characters of variable length eg  * Lim Ah Seng,123 Sesame Street * Joy Tan,10 Rhu Ave * this optimises storage compared to fixed length record fields where maximum space needs to be allocated for the longest string * most computing system have built-in facilities to read and extract the individual fields separated by the delimiter character |
| 1. (Is this correct?) A continuous string containing all attributes of the record, followed by an array containing the length of each attribute for the particular record    * e.g. Lim Ah Seng123 Sesame StreetJoy Tan10 Rhu Ave, [11,17,7,10] 2. a **preceding** line indicating the **length of each attribute** (separated by space character), followed by the actually record line eg  * 11 17 7 10 * Lim Ah Seng123 Sesame StreetJoy Tan10 Rhu Ave * program will read and extract the relevant attribute based on its length provided |
| 1. <http://en.wikipedia.org/wiki/Tab-separated_values>    * preferred to .csvs & other delimiter-separated variable formats when the data in question would be expected to contain characters commonly used for delimiters (e.g. comma, semicolon, tilde) |

**(d)** Using one of the methods described in part (c) give an algorithm for finding the nth   
 record in a serial file. (Assume that the file starts with record 1 and only serial access   
 is allowed.) [4]

|  |
| --- |
| // using 2 - preceding line contains attribute lengths  // assume first attribute is key / comparison attribute  OPEN serial file in read mode  REPEAT n \* 2 TIMES  read attribute length line  record\_length = sum of all attribute lengths (loop)  read record line using record length  OUTPUT nth record line  CLOSE serial file |

**(e)** Data held in two sequential files is to be merged into a new sequential file.

**(i)** State **two** conditions for this merge process to be possible. [2]

|  |
| --- |
| 1. Both files must have the same **record structure** 2. Both files must be sorted in the **same order** ~~i.e. both either ascending or descending~~ according to comparison attribute |

**(ii)** Describe the algorithm to carry out the merge. [4]

|  |
| --- |
| OPEN file1, file2 for READING/INPUT  CREATE newfile for WRITING/OUTPUT  item1 = READ item from file1  item2 = READ item from file2  WHILE item1 not EOF(file1) OR item2 not EOF(file2)  # run WHILE loop as long as either file not completely  # traversed  IF item2 == EndOfFile or item1 < item2:  # if end of file2 reached, loop until end of file1  WRITE item1 to newfile  item1 = READ item from file1  ELSE:  WRITE item2 to newfile  item2 = READ item from file2  CLOSE all files |

**2012P1**

**6** An Abstract Data Type (ADT) consists of both a data type and its associated operations.

A linked list ADT has the following operations defined:

|  |  |
| --- | --- |
| **Operation** | **Description** |
| Create(L) | creates an empty linked list L |
| Insert(L, item, P) | insert new value, Item into linked list L so that it is at position P in the linked list |
| Retrieve(L, P) | get a copy of the item at position P in the linked list L |
| Remove(L, P) | remove the item at position P in the linked list L |
| IsEmptyList(L) | returns true if linked list L is empty |
| Size(L) | returns the number of items in the linked list L |

**(a)** The following operations are to be carried out:

1. Create(L)
2. Insert(L, Pre, 1)
3. Insert(L, Red, 1)
4. Insert(L, Pit, 2)
5. Retrieve(L, 1)
6. Insert(L, Abe, 2)
7. Retrieve(L, 3)
8. Remove(L, 2)
9. Insert(L, Ben, 3)

Describe the linked list after

**(i)** The first five operations have been carried out; [3]

|  |
| --- |
| 1. A linked list called ‘L’ containing three items would have been created. 2. The start pointer would point to the item ‘Pre’ in position 1; 3. The item ‘Red’ is then inserted into the linked list in position 1, causing the item ‘Pre’ to occupy position 2. 4. The third item ‘Pit’ is then inserted into the linked list at position 2, causing the item ‘Pre’ to occupy position 3. 5. The first item, ‘Red’ is then retrieved.   The linked list L is created with 3 items inserted in the order: Red, Pit, Pre.  The item at position 1 i.e. Red is then retrieved. |

**(ii)** The final four operations have been carried out. [1]

|  |
| --- |
| After inserting Abe at the second position, linked list L contains Red, Abe, Pit, Pre.  The item at position 3 i.e. Pit is then retrieved.  The item at position 2 i.e. Abe is then removed.  The item Ben is then inserted at position 3.  Thus the linked list L contains the items in the order: Red, Pit, Ben, Pre. |

**(b)** A linked list, ORDERED, contains integers that are stored in ascending numerical   
 order. The linked list has a null terminator.

Write pseudocode, using the operations given above, which would place an item,   
 NewItem, in its correct place in the linked list. Allow for all possibilities. [7]

|  |
| --- |
| Create(ORDERED) // if not already exist  IF IsEmptyList(ORDERED) // insert as first item  Insert(ORDERED, NewItem, 1)  ELSE // look for correct position to insert  CurrentItem = Retrieve(ORDERED, 1)  i = 2  WHILE (NewItem < CurrentItem) AND CurrentItem != null  CurrentItem = Retrieve(ORDERED, i)  i = i + 1  IF NewItem == CurrentItem // duplicate  output duplicate value error  ELSE  IF CurrentItem == null // insert as last item  Insert(ORDERED, NewItem, i)  ELSE // insert to correct position  Insert(ORDERED, NewItem, i - 1) |

**(c)** Describe a queue data structure. [4]

|  |
| --- |
| A Queue data structure is a data structure which collects data following a **First-In-First Out and Last-In-Last-Out (FIFO/LILO) format**, meaning that an Insert function would see data be **inserted at one end** (rear) and a Remove function would see data **deleted from the opposite end** (front), much like how a queue in real life would work. |

**(d)** A queue is also an ADT.

Define the Insert and Remove operations for the queue, Q. [4]

|  |  |
| --- | --- |
| Insert(Q, item) | insert new value, item as the last/rear item in queue Q |
| Remove(Q) | remove first item from the front of queue Q |

**(e)** For the operations given in **(d)** show how they could be implemented using the linked   
 list ADT operations. [4]

|  |
| --- |
| // assume Q already created / exists  Insert(Q, item)  Insert(Q, item, Size(Q)+1)  Remove(Q)  IF IsEmptyList(Q)  output cannot remove from empty queue error  ELSE  Remove(Q, 1) |

**(f)** Give **one** advantage and **one** disadvantage of using an array data structure when   
 implementing the linked list ADT. [2]

|  |
| --- |
| * Advantage:using an array data structure will allow ~~easy~~ **direct/random access** of data from **any position** in the linked list ADT using the array index/subscript * Disadvantage: **reorganisation** will be required every time a new integer is added to the front or middle of the linked list, making it **less efficient**; also need to allocate **contiguous memory space** which will be suboptimal if storage requirements are not well known in advance |

**2012P2**

The transport ministry is responsible for all major roads in the country. This responsibility includes the repair of road surfaces.

**1** When a road surface is being repaired it is sometimes necessary to use temporary lights   
 to control the traffic. These temporary lights operate a set, timed sequence.

The ministry has decided that traffic control would be improved by using   
 computer-controlled lights which would change according to traffic conditions.

**(a)** A systems analyst is employed to solve the problem of controlling the lights. Explain   
 the importance of precisely formulating the problem to be solved and the way that the   
 ministry representatives and the analyst contribute to the problem definition. [3]

|  |
| --- |
| Precisely formulating the problem involves:   * setting the scope of the problem such as an **accurate understanding** of the situation: problems encountered during repairing of roads and how temporary lights should solve each of them   + Ministry representations can identify the timeline, location and traffic situation of the different roads that will be repaired. Problem solution and resource allocation will be designed based upon the varying situations.  ie. road is in CBD, has generally heavy traffic, hence might require more traffic lights, against a road in the countryside. * **efficient allocation** of scarce resources, with more allocated towards higher priority problems allocation of resources include the budget provided, the distribution of labour/workload for different personnels involved (ministry representatives, analyst, construction workers, etc). higher priority problems such as implementation of the traffic lights should have more budget/personnels allocated towards them   + Analysts, base on the different problem situations provided by the Ministry Representations, will allocate resources accordingly for each road repair. ie. a road in CBD will require more resources allocated there, such as budget, traffic lights and maybe even traffic marshals, against a road in the countryside. |

**(b)** Specify the input and output requirements of the system to control the traffic. [3]

|  |
| --- |
| Input requires **hardware** (such as proximity sensor, light sensor, video camera) that detects the traffic situation (light or heavy) on the road.  Input requires certain **software processing** to process the sensor input and identify traffic situation.  Output requires **hardware** that relays traffic control orders to drivers effectively.  (traffic lights, visual relay: red light to stop traffic, green light to let traffic flow) |

**(c) (i)** It is decided to use a light sensor to detect the presence of a waiting vehicle.

Describe how the light sensor is used and give a disadvantage of using this type   
 of sensor as an input device for this application. [2]

|  |
| --- |
| * In the event there are no cars around, the light sensor would detect light from sources such as the Sun or a nearby lamppost. When there is an awaiting vehicle nearby, the light sensor will be **obstructed** by it. The non-presence of light detected by the sensor would therefore imply a waiting vehicle. * One possible disadvantage of using the light sensor would be that in scenarios such as stormy days or faulty lampposts, the light sensor would give an **inaccurate reading** of the surroundings |

**(ii)** State another type of sensor which would be a more appropriate input device,   
 justifying your answer. [2]

|  |
| --- |
| * Proximity Sensor * A proximity sensor, detecting the presence of waiting cars by calculating the distance between itself and nearby large objects, would work regardless of the weather or uptime of surrounding light-emitting devices. Therefore given the context of application, a proximity sensor would be a more appropriate input device than a light sensor |

**(d)** When the software for controlling the lights is tested, both black and white box testing   
 are used. Both syntax and logic errors are found.

State what is meant by:

**(i)** black box testing [1]

|  |
| --- |
| Functional testing from the perspective of the end-user, to ensure that the program output, given certain input, would **conform to the specified functions**.  ie. When presented with a range of traffic situations (heavy/light), traffic lights should yield its proper traffic controls. |

**(ii)** white box testing [1]

|  |
| --- |
| **Structural testing** carried out by **programmers** to ensure that the internal structure of the program is sound; considers **all possible outcomes conceivable** within the program and ensures each and every possibility is accounted for and working.  ie. Within the program, programmers can directly generate all possible/exhaustive permutation of traffic conditions as input, to ensure all control paths in the module are exercised, including error conditions such as traffic light malfunction. |

**(iii)** a syntax error [1]

|  |
| --- |
| Structural error of the program, where a part of the code has violated the **grammar/rules** of the program language.  ie. Assuming code is written in python  if traffic\_condition > 500: //500 being a placeholder number indicating heavy traffic  retrn REDLIGHT  // ‘return’ is spelled wrongly, hence program yields a grammatical/syntax error. |

**(iv)** a logic error [1]

|  |
| --- |
| An error where the program **runs successfully**, but produces an **undesired/invalid/incorrect** result.  ie. A heavy traffic condition should yield a red light traffic control output, however green light is the output instead. Program functions, but produces an incorrect result; logic error occurs. |

**2012P2**

**2** Cameras are placed at fixed regular intervals on the roads. Each takes an image of the   
 registration plate of a vehicle that passes. This image is stored together with the time that it   
 was taken.

**(a) (i)** Explain why some citizens are concerned about these data being captured.

|  |
| --- |
| // answer leading to the next part  The image taken may have **sensitive information** such as the faces of the people in the car, which may be viewed as an **intrusion of privacy**.  // alternative answer  The data contains the specific time and location of every car, which reveals the location of the car at any point in time.  Such data is considered **sensitive**, allowing tracking of vehicles’ whereabouts, which is an **invasion of privacy**, allowing parties (such as the government, or hackers) to trace them **without their consent**. |

**(ii)** Describe steps that can be taken to respond to their concerns.

[5]

|  |
| --- |
| An automatic face recognition system could be employed to allow **real-time censoring** of photos such that faces are not exposed by the camera. The face recognition system will identify faces with a certain margin of error given, and apply either a blurring effect or covering the face with a black square. This will prevent the people viewing the data from being able to know who is in the car and hence **protect the identity of the people in the car**.  //answer for alternative  Ensure that the data is secure from third-parties or unauthorised users who may misuse the information retrieved from the data. Perhaps password-protection may be employed to grant permission only to authorised parties.   * ensure all data captured is encrypted and transmission is over secure channel * install firewall and anti-malware program to detect unauthorized access and intrusion * have data protection act/policy that ensures users all data captured are only used for its intended purpose and will be deleted after a reasonable period of time eg 3 months * have laws to enforce adherence and institute serious impediments/punishments eg hefty fines or jail terms |

**(b)** When a vehicle's registration code is captured, a search is made of the file of   
 registration codes currently stored. If it is not present in the file it is appended to the   
 end of the file.  
 The file is called REGISTRATIONS.

**(i)** State the name for this type of file organization. [1]

|  |
| --- |
| Serial File Organization (as it is ordered mostly by date and time of data entry) |

**(ii)** Describe the process of searching this file for a particular registration code and   
 the addition of the registration code to the file if not present. [5]

|  |
| --- |
| * Process of searching a serial file, which is organised in chronological order, can be done via a **linear search method** applied to the system. * The system will **iterate through** the serial file containing the registration codes and check whether the registration code is contained within the file. * If the registration code is found inside the file, the particular registration code is present and the program returns true. * When the program reaches the end of the file and the registration code is not found, this implies that the registration code is not present in the file and * the program will write a new line in the file at the end of the file containing the registration code. |

**(c)** An algorithm is written to process data from a file. Each record in the file is of fixed   
 length comprising three fields. The first field is a registration code (R) which is the   
 primary key.

The other two fields are T1 and T2, which are the earlier and later times at which the   
 vehicle passed the previous two cameras respectively.

The time (T) is the time when the vehicle passes the current camera.

A value of NULL means that there is no time recorded.

INPUT R, T

SEARCH(R)

IF T1 = NULL THEN T1 = T

ELSE IF T2 = NULL THEN T2 = T

ELSE T1 = T2, T2 = T

IF T2 <> NULL THEN X = T2 - T1

IF X < 10 THEN OUTPUT R

**(i)** State **three** different techniques which should have been used when the   
 algorithm was produced to make it more understandable to the reader. [3]

|  |
| --- |
| 1. Use of **meaningful identifier names** instead of R,T so that other people maintaining the code can tell what variable it is from the first glance. 2. Using multiple lines to state the if/end-if conditions instead of condensing both the setting of variables and the conditions for the if-statement in one line, along with **indentations and sensible use of white space** allowing for end statements to be expressed for **clarity** as well. 3. Use of appropriate **annotation/comments to aid the flow of logic** through the program or explain difficult/cryptic parts. |

**(ii)** Apply these techniques by rewriting the code above. [3]

|  |
| --- |
| INPUT RECORD, TIME  SEARCH(RECORD)// Obtain TIME1 and TIME2, and/or update them.  IF TIME1 == NULL // No TIME1, set TIME as TIME1  TIME1 = TIME  ELSE IF TIME2 == NULL  // No TIME2, TIME1 present, set TIME as TIME2  TIME2 = TIME  ELSE  // Both present, replace TIME1 with TIME2  // and replace TIME2 with TIME  TIME1 = TIME2  TIME2 = TIME  ENDIF  IF TIME2 <> NULL  // If there are two times, compute the difference  DIFFERENCE = TIME2 - TIME1  IF DIFFERENCE < 10  // Output record if difference is less than 10  // This denotes that the car is speeding  OUTPUT RECORD  ENDIF  ENDIF |

**(iii)** Explain what the algorithm does. [6]

|  |
| --- |
| A registration code and the time that the vehicle passes the current camera are input.  The algorithm checks if the vehicle passed a camera previously. If it did not, the earlier time of it passing a camera is set to the current time. If the vehicle passed a camera previously, but did not pass a second camera, the later time is set to the current time.  If the vehicle passed two cameras before, the time it passed the first camera is set to the time it passed the second camera, and the time it passed the second camera is set to the current time.  If the vehicle has passed two cameras, the time difference between passing camera 1 and camera 2 is found by subtracting the time it passed camera 1 from the time it passed camera 2. If this time difference is lower than 10 seconds, the vehicle is assumed to be speeding and its registration details are output to the file of registration codes.   * It searches the records in the file until the record with the same registration code as RECORD is found. * The variables TIME1 and TIME2 are then returned from the data. * If TIME1 is found to not exist, the input TIME will overwrite TIME1. * If TIME1 exists, but not TIME2, TIME2 will be set to TIME. * After that, the difference between TIME2 and TIME1 is checked if both are present. * If the difference is less than 10, the record is given as output to the user. |

**(d)** A registration that is output from the algorithm in part **(c)** is stored in a file called   
 SPEEDING. Details of vehicles are stored in another file called VEHICLE. Each   
 vehicle's registration code may occur more than once in the SPEEDING file, but each   
 record in the SPEEDING file relates to only one vehicle. Each vehicle in the VEHICLE   
 file is owned by one owner whose details are kept in the OWNER file. Each owner   
 may own more than one vehicle.

**(i)** Draw an ER diagram to show how these files are related. [3]

|  |
| --- |
| Each owner may own more than one vehicle.  Each vehicle may have many records in the SPEEDING file. |

**(ii)** For each record in the SPEEDING file, a letter is to be sent to the owner of the   
 vehicle.

Explain how the contents of the files can be used to send letters to these owners. [4]

|  |
| --- |
| * letter will contain information such as vehicle registration code, owner name and address, speed recorded and data and time of offence * these data are **stored in different/separate tables** in the optimized/normalized database design, and hence need to be retrieved using SELECT query JOIN statements with the link/foreign keys as the criteria * eg SELECT vehicle\_registration\_code, owner\_name, owner\_address, speed, date\_time\_recorded FROM SPEEDING, VEHICLE, OWNER WHERE SPEEDING.vehicle\_registration\_code = VEHICLE.registration\_code AND VEHICLE.owner\_name = OWNER.name * a word processor with mail merge feature can then **capture the fields from the view generated by the query** to **automatically populate** a letter template |

**2012P2**

**3** Vehicles of different types can be assigned to different classes.



**(a) (i)** Explain the term 'inheritance' by using an example from this class diagram. [2]

|  |
| --- |
| * Ability to **adopt data/properties and methods/operations** from its parent/superclass (hence promoting **code reuse**) and also to **define its own data and methods** (hence allowing specialisation) * Truck is a subclass of the Vehicle superclass and hence adopts all properties (eg registration code) and methods of its parent class. It can also define its own properties and methods specific to itself (eg axles and company).   Inheritance refers to the ability of a class to adopt the data and methods used by a superclass, then allowing it to add new data and methods. For example, the class Car inherits the data registration code from Vehicle, and is also able to define new data make and model. |

**(ii)** A Jowett Javelin is a car.

Explain why its properties include registration code but not company. [3]

|  |
| --- |
| * Jowett Javelin is an instance/object of class Car and hence has **all properties associated with the Car class** as well as properties **inherited from its superclasses** * registration code **inherited from superclass Vehicle** as it is a common and required identification attribute across all types of vehicles * car is private vehicle owned by individuals and hence **company is not an essential attribute of cars** but for trucks |

**(iii)** Motorbikes are to be included in the diagram. The property EngineSize should be

included, as should the method GetEngineSize().

Draw the class Motorbike and state where it should fit on the class diagram   
 above. [4]

|  |
| --- |
| * subclass of Vehicle (upward pointing arrow indicating inheritance) * class name Motorbike * with property EngineSize * and method GetEngineSize() * and Display() method (to show polymorphism) |

**(b)** Use the information given to explain the terms.

**(i)** class

|  |
| --- |
| * a class is a **blueprint/template** for **bundling/grouping together** the **common** **data/attributes/characteristics and methods/operations/behaviour** of an entity we wish to model using abstraction of the problem domain * Vehicle is a class and contains data registration code and constructor method to allocate storage and create object, accessor method get\_registration\_code() to read/provide the value of registration code to the main program or other classes   A class is a set of data and methods used as a template to create objects that possess these data and methods. |

**(ii)** object.

[4]

|  |
| --- |
| * an object is **an instance/instantiation of a class** at program runtime. Each object has its **own state space** but **shares the same method space** as other objects of the same class. * car1 is the name of the object of the Car class and will contain its specific instance values for registration code, make and model, together with the ability to invoke its own constructor, accessor, modifier/mutator and helper/support methods   An object is an instance of a class, where it possesses the data and methods defined by the class. |

**2012P2**

**4** Details of people who own vehicles are stored in the transport ministry in a file called   
 OWNER.

**(a)** State the data types that would be used to store data in the following fields in the file:

**(i)** Amount of money owed in traffic fines.

|  |
| --- |
| ~~Integer~~  < Incorrect, money requires decimal places!!  Unless justified by storing in terms of cents.  Float/Real - for storing of numeric data **with decimal places**, which money requires.  For **arithmetic operations** to be carried out   * addition: when owner is fined * subtraction: when owner pays fines |

**(ii)** Type of licence, coded as a single letter:

* F - fill licence issued one year after passing the driving test;
* P - provisional licence issued to learner drivers before passing the driving test;
* R - recently passed test.  
   [2]

|  |
| --- |
| Character - single letter indicating type of license |

**(b)** The name of the owner of a vehicle is stored as a string of characters represented by   
 ASCII codes.

Explain what is meant by ASCII codes. [3]

|  |
| --- |
| ASCII stands for **American Standard Code for Information Interchange**, a **7-bit** character encoding scheme with 128 specified characters based on the English language. Characters include **printable characters** like alphabets (lower and uppercase), numerals, special characters like \*&^%, and **non-printable control characters** like backspace, carriage return |

**(c)** Data stored about an owner may need to be changed and new owners may need to   
 be added. These changes to the contents of the file are added by keyboard   
 operators. The operators may be presented with details of changes on a printed form   
 or they may speak directly to an owner when the owner telephones to change their   
 details.

Describe an appropriate user interface for this data entry application, justifying your   
 choice. [4]

|  |
| --- |
| A graphical user interface (GUI). The (GUI) allows the operators to scroll through the file and edit the data stored about the owners easily by **providing graphical icons and/or visual cues** that allow users to input data using the keyboard and the mouse. The keyboard operators **do not need in-depth knowledge on the requisite workings of the system** as the visual cues help them to manipulate the application more easily **without the need for extensive documentation** or any documentation at all.  Also, a GUI, using the manipulation of graphical elements, allows room for trial and error by the keyboard operators, with a much greater error margin as compared to a command-line interface. |

**(d)** Communications that come into the ministry are concerned with vehicles or owners. A   
 vehicle may be new or may have been scrapped, alternatively an existing vehicle   
 may have changed owner. Owners may be applying for a different type of licence or   
 may be paying money owed in fines.

Draw a decision tree which can be used to deal with these communications, showing   
 suitable outcomes. [8]

|  |
| --- |
|  |

**2012P2**

**5** The transport ministry is responsible for all the city car parks.

**(a)** At each car park:

A car arriving triggers a sensor (S1) and a fixed fee (F) is paid into a machine. This   
 allows a barrier (B1) to be lifted and the car to enter the car park. When a car leaves   
 the car park it passes over another sensor (S2) and another barrier (B2) is lifted.

Each car park has a maximum number of spaces for cars (M) and when this maximum   
 is reached a 'FULL' sign is illuminated at the entrance and the barrier (B1) will not rise.   
 The car park is closed at least once a day for cleaning purposes.

Write an algorithm which will control the barriers and which will keep a total (T) of the   
 fees paid. [8]

|  |
| --- |
| //initialization  Integer CONSTANT M = max number of cars  Integer N = 0 //current number of cars  Float CONSTANT F = fixed fee.  Float T = 0 //total amount of money in machine  Boolean Cleaning = False //default to be false  Sensor Object S1,S2  Barrier Object B1,B2  //main  Input cleaning  while not Cleaning:  if N == M then :  B1.illuminate(“FULL”)  if S1.triggered() and N<M then:  B1.activate()  T = T + F  N = N + 1  if S2.triggered() then:  B2.activate()  N = N - 1  if N == 0 then : // if no car in carpark then can start cleaning  INPUT cleaning //can change cleaning state to true if necessary  endwhile |

**(b)** There are 100 car parks, each of which is identified by a number between 1 and 100.

At the end of each month the total fees paid for that month (T) is collected from each   
 of the car parks as an integer value.

All data are stored in an array Parks().

* For car park x, Parks(x,1) to Parks(x,12) contain the totals for the twelve months of the year.
* Parks(x,13) contain the annual total fees collected for each car park.

Using Parks(x,y) to identify individual values in the array, write an algorithm which   
 can be used to produce the annual totals once the twelve monthly totals have been   
 input to the array. [5]

|  |
| --- |
| FOR X FROM 0 to 99  BOOLEAN FILLED  SET FILLED TO TRUE  FOR Y FROM 0 TO 11  IF Parks(x,y) = NULL //ROW Y IS EMPTY  FILLED = FALSE  OUTPUT INCOMPLETE ROW Y ERROR  BREAK  ENDIF  ENDFOR  IF FILLED  TOTAL = 0  FOR Y FROM 0 TO 11  TOTAL = TOTAL + Parks(x,y)  ENDFOR  Parks(x,13) = TOTAL  OUTPUT TOTAL  ENDIF  ENDFOR |

**2011P1**

**4** The following diagram show the syntax of a statement, a variable, a letter and an   
 operator in a particular programming language.

**~~(a)~~** ~~State whether each of the following are valid statements or not. Give a reason for your answer in   
 each case.~~

**~~(i)~~** ~~A = B + C;~~

**~~(ii)~~** ~~A := B - 2;~~

**~~(iii)~~** ~~ABCDE := AC + AD~~

~~[6]~~

**~~(b)~~** ~~Write the above syntax diagrams in Backus Naur Form (BNF). [6]~~

**(c)** Code is to be written that will check whether a variable is valid or not. The input will   
 be a string of characters (including spaces).

Give an algorithm for a recursive function which would return a value of ‘true’ if the   
 variable is valid but ‘false’ if it is not valid. [6]

|  |
| --- |
|  |

**2011P1**

**6** The following words are to be placed in a binary tree data structure:

rice curry sugar salt coffee tea ghee

**(a)** Draw the alphabetically ordered binary tree after all the words have been added in the   
 order given above. . [6]

|  |
| --- |
|  |

**(b)** Write down the output if the tree is traversed using [6]

**(i)** In-order traversal

|  |
| --- |
| Coffee Curry Ghee Rice Salt Sugar Tea |

**(ii)** Pre-order traversal

|  |
| --- |
| Rice Curry Coffee Ghee Sugar Salt Tea |

**(iii)** Post-order traversal

|  |
| --- |
| Coffee Ghee Curry Salt Tea Sugar Rice |

**(c)** Write an algorithm for the insertion of a word into an alphabetically ordered binary tree.   
 Assume that the word does not appear in the tree already. [5]

|  |
| --- |
| function Insert(BST, word)  if initialData > word  if dataLeft is None  dataLeft -> BST(word)  else  dataLeft.Insert(word)  else //if self.data < data  if dataRight is None  dataRight -> BST(word)  else  dataRight.Insert(word) |

A binary tree abstract data type (ADT) is to be implemented by using a class where   
 each object is a node of the binary tree.

**(d)** Describe **three** properties of the class. [6]

|  |
| --- |
| data: value stored in each node  left: root of its left subtree  right: root of its right subtree |

**(e)** State **two** methods that would be required for the ADT. [2]

|  |
| --- |
| Create(BST) - create empty/new binary search tree  Insert(BST, item) - insert new node with item value to binary search tree  Search(BST, target) - search for target value in binary search tree  Inorder(BST) - traverse and output contents of binary search tree in ascending / alphabetical order |

**2011P2**

A company called Fish4T specialises in the sale and care of tropical fish.

The company has three retail shops and a warehouse as well as a separate office building. The shops sell fish, fish tanks and the equipment for the tanks as well as fish food.

**1 (a) (i)** The shops feature information systems with touch screens that allow customers to   
 find out about fish and other products available.

Describe a user interface which would be sensible for user with these screens,   
 justifying your choice. [4]

|  |
| --- |
| menu-based interface? |

**(ii)** Fish4T also run a website which allows users to place orders for goods

Describe a user interface which would be sensible for use with the website to allow   
 users to order goods, justifying your choice. [4]

|  |
| --- |
| form-based interface? |

**(b)** When data is input on the website it is stored as ASCll codes representing the   
 character set.   
 Explain what is meant by

**(i)** the character set of a computer. [3]

|  |
| --- |
| * well-defined **collection of allowable** (printable and non-printable/control) **characters** and symbols * for the **entry (input) and display (output)** of textual information * usually tied to a particular written language (eg English) |

**(ii)** ASCll codes. [3]

|  |
| --- |
| * **American Standard Code for Information Interchange**, a **7-bit** character encoding scheme with 128 specified characters based on the English language. * Characters include **printable characters** like alphabets (lower and uppercase), numerals, special characters like \*&^%, * and **non-printable control characters** like backspace, carriage return |

**(c)** When the amount of money spent on an order is stored in the computer system, it is   
 stored as a numeric variable.

Explain how it can be represented in **two** different ways. [2]

|  |
| --- |
| * **real/float/double data type with fractional parts** eg 23.50, ~~mantissa and exponent are stored separately eg 23.50 (=0.235 \* 10~~~~2~~~~) will be stored using .235 as mantissa and 2 as exponent (both converted to binary and normalized)~~, will contain **truncated values** as some numbers **cannot be accurately stored using the finite number of bits** in a computer system. * **integer data type** with monetary values stored to the nearest cents eg $23.50 will be stored as 2350, ensures **accurate representation of numbers** at the expense of reduced range of values stored |

**2011P2**

**2** The information about the fish stocked by Fish4T is stored in a sequential file using the   
 names of the different types of fish as key.

**(a)** Write an algorithm that will perform an efficient sequential search on the file to find   
 information about a type of fish. [8]

|  |
| --- |
| idea - early termination when found or when target exceeds current fish name (not found); sequential files are ordered by their keys, so records are already sorted, so just need to compare keys to find the information. Use binary search to find correct key.  Must include:   * Meaningful variable names * Comments/Annotations * Adequate use of whitespace |

**(b)** If the file stored information about all tropical fish, rather than just the types stocked by   
 Fish4T then a different type of search algorithm may be used.

Describe, briefly, two other forms of search algorithm. Give a reason why each would   
 be appropriate in this case in preference to a sequential algorithm. [6]

|  |
| --- |
| Sequential search has longest runtime of O(n)  random file - hash table search   * Provides direct access to the information since we already have the key. * Longest runtime of O(1) * More efficient, faster data retrieval than sequential searching.   binary search tree - binary search   * Splits the problem into two during each search and since our data is already sorted by key, we will have easy retrieval * Longest runtime of O(log2(n)), where n is the size of the data * More efficient, faster data retrieval |

**(c) (i)** The data about sales in each shop is collected and processed at the end of each   
 day.

State the type of file organization used to collect the sales date during the day.   
 Explain how it differs from the way that the information about the fish is organized. [3]

|  |
| --- |
| Serial file organisation.  Serial file is in chronological order while sequential file is in index order. |

**(ii)** Each time a new sale is made, a record is added to the sales file in the shop

Each time a new type of fish is stocked by Fish4T a record is added to the file of   
 stocked fish.   
 Briefly describe the methods used to add the new records to each of these two files. [4]

|  |
| --- |
| In a serial file, each new record is added to the back of the file since the file is chronologically ordered. To add a new record to the sales file, just append the record to the end of the sales file.  In a sequential file, the records are arranged by key order. To add a new fish record to the file of stocked fish, STOCKS, copy all records to a new file, NEWFILE, until the key is larger than the new fish key. Then, write the new key and record into NEWFILE. Copy the rest of the file to NEWFILE before replacing STOCKS with NEWFILE. |

**2011P2**

**4** The class diagram shows some of the information held about fish on the Fish4T system.

**(a)** Using this diagram explain what is meant by inheritance. [3]

|  |
| --- |
| Inheritance is the adoption of data/attributes and methods from the superclass by subclasses such that subclasses are also able to use the same attribute names and methods. For example, [Subclass] also has attributes [attribute1] and [attribute2]. It may also invoke the methods [method1] and [method2] as it has inherited these attributes and methods from [Superclass].  Subclasses may also define their own methods and attributes. [Subclass] has the attribute [attribute3] which is unique only to [Subclass] and it may invoke the method [method3] which [Subclass2] and [Superclass] do not have access to.  \*\* Make reference to the diagram. |

**(b)** ‘Snails’ should be included in the class diagram. One of the attributes will be   
 Watertype and one of the operations will be getWatertype().

Show this information on a copy of the diagram and say how it would fit in the class   
 diagram above. [4]

|  |
| --- |
|  |

**(c)** A ‘Tiger Snail’ is a new creature to be included. The following code may be used:

oneSnail = Snail new

oneSnail.getWatertype(‘Fresh’)

Use the information given to explain, and give an example of, each of these terms:

**(i)** object [3]

|  |
| --- |
| An object is an instance of the class and has methods and data that are specified by the class. The object oneSnail is an instance of the class Snail and is allowed to use methods such as getWatertype() that were defined in the class. |

**(ii)** class [3]

|  |
| --- |
| A class is a template/blueprint from which instances of the class are to be based on. It is an abstraction of data and methods to model real-life entities. All instances have the same data and methods that are specified in the class, such as all instances of Snail have the method getWatertype(). |

**2011P2**

**~~5 (a)~~** ~~Each point of sale terminal has a number of input and output devices.~~

**~~(i)~~** ~~State~~ **~~two~~** ~~input devices and explain why they would be necessary. [4]~~

**~~(ii)~~** ~~State~~ **~~two~~** ~~output devices and explain why they would necessary. [4]~~

**(b)** The water in a fish tank must be kept at a suitable temperature for the fish that are in   
 it. A processor is used to accept an input from a temperature sensor and to produce   
 an appropriate output to a heater if the water is too cool. It may be assumed that the   
 water does not need a device to cool it if it gets too hot, simply that the heater should   
 be turned off. A warning light should be turned on if the temperature is out of the   
 range 24-26 degrees.

The fish in a particular tank like the water at 25 degrees.

Using T to stand for the temperature reported by the sensor and H for the heater   
 which can either be 0 (off) or 1 (on), write an algorithm in pseudocode to control the   
 temperature of the water. [8]

|  |
| --- |
| integer T = sensor.getTemperature()  boolean H = 0 // set to off  if T < 25  // switch on heater  H = 1  if T < 24  warningLight.switchOn()  else if T > 25  // switch off the heater  H = 0  if T > 26  warningLight.switchOn() |

**2011P2**

**6** Telephone orders to the company are taken by operators in the offices.

The orders are stored on a file and then sent to the warehouse for processing.

The order is also sent to the accounts department so that an invoice can be sent to the   
 customer.

A copy of the invoice is kept for future reference.

**(a)** Draw a data flow diagram (DFD) to illustrate this process. [6]

|  |
| --- |
| Entities: Customer, Operator, Accounting Department, Warehouse  Processes: Store data in file, send to accounting department, send to warehouse, produce invoice  Data Store: Orders |

**(b)** An analyst is employed to oversee the production of a new program to control the   
 ordering system.  
 It is decided to use a top-down design and modular approach to the programming.

Explain this approach. [4]

|  |
| --- |
| Top-down design refers to building the components of the programme from highest priority to lowest priority. Subprograms or routines that are not as crucial to the programme’s functionality are developed last, which allows most features of the programme to be available for testing at an early stage.  Modular approach means developing the programme module by module. It requires splitting up of the programme into simpler functions and routines, which are then developed independently from the other functions. This allows the different functions in the programme to be able to function perfectly on their own. |

**(c)** When a module has been written, it may contain errors.

Describe each of the following types of programming error. Give an example and   
 state how it will be discovered.

**(i)** Syntax error . [3]

|  |
| --- |
| Syntax errors arise when the compiler receives commands that do not adhere to the grammar rules of the specific programming language. For example, the line “if test = True” will produce an error when compiled in Python language. It should have been given as “if test == True” instead. The error will be flagged by the compiler, providing the programmer with the line at which the error occurred. |

**(ii)** Logic error [3]

|  |
| --- |
| Logic errors arise when the commands are executed successfully, but the programme does not give the intended output. For example, when computing the value of area of a circle, the command “area = pi \* radius \* radius \* radius” would produce a logic error. The correct command should be “area = pi \* radius \* radius”. This error cannot be detected by the compiler and may only be discovered by matching programme outputs with expected outputs. |

**(d)** All the modules have been thoroughly tested and work perfectly.

Explain why the full program does not work when the modules are combined. [2]

|  |
| --- |
|  |

**2010P1**

**2** Fig. 1 shows three classes and some of their private attributes and public methods.

(uml diagram)

**(a)** Using examples from Fig. 1, explain the following terms:

**(i)** encapsulation, [3]

|  |
| --- |
| Encapsulation refers to the bundling of private data and public methods within a class. Private data may only be accessed by public methods (interface), which prevents external code from interfering/modifying data values in the class.  lol  For example, the data [data1] and [data2] from [Subclass 1] may not be accessible by other classes such as [Subclass 2] since it is encapsulated in [Subclass 1]. Only by using [method 1] and [method 2] on [Subclass 1] may [data1] and [data2] be retrieved or modified. |

**(ii)** inheritance, [3]

|  |
| --- |
| Subclasses adopt attributes and methods that have been previously defined by the superclass, promoting code reuse. Subclasses may also define its own attributes and methods and override previously-defined methods.  For example, ... |

**(iii)** polymorphism. [3]

|  |
| --- |
| Polymorphism is the ability to create an object, variable or method that has more than one form. It allows for different methods of the same name to be invoked. Dynamic binding occurs at program runtime.  For example, [method1] from [Subclass] has the same name as [method2] from [Superclass]. However, [method1] is a different method from [method2] and the programme calls the corresponding method during runtime. |

**(b)** Explain what is meant by data hiding. [2]

|  |
| --- |
| The ability of an object to have access to only the methods and attributes that are related to the object. Thus, external methods that are not native to the object may not be used to access the data defined in the object. |

**(c)** For each of the following classes give one other possible private attribute:

**(i)** SalesStaff,

|  |
| --- |
| sales\_figures |

**(ii)** TemporaryStaff. [2]

|  |
| --- |
| contract\_period |

Some permanent staff earn commission on each sale that they make. This commission   
 is a percentage of the value of each sale. However, the greater the total value of the   
 sales that they make the higher the percentage used.

**(d)** Describe two ways that Fig. 1 could be altered to accommodate this additional   
 information. [7]

|  |
| --- |
|  |

**(e)** Give **two** advantages of using an object-oriented programming language. [2]

|  |
| --- |
| * Promotes code reuse   + Many instances of the same class may be produced, but since all of them make use of the same methods and attributes, we do not have to redefine them separately. * Decomposing a huge problem into smaller portions that can be solved separately.   + Methods are independent of each other, so developing the solution would be more manageable. * Abstraction that can be easily understood by humans   + Individual modules and their relationships with the classes can be easily understood * More: <http://www.felixgers.de/teaching/oop/oop_intro.html> |

**2010P1**

**4** An Abstract Data Type (ADT) consists of both data type and associated operations.

A stack ADT has the following operations defined:

(i) Create(S) - creates an empty stack S,

(ii) Insert(S, Item) - insert new value, Item, onto stack S,

(iii) Retrieve(S) - retrieve item from the stack, S,

(iv) EmptyStack(S) - returns true if stack S is empty.

**(a)** Describe the stacks after the following operations have been carried out:

Create(S1)

Create(S2)

Insert(S1, London)

Insert(S2, Paris)

Insert(S1, Rome)

Insert(S2, Lisbon)

Retrieve(S2)

|  |
| --- |
| * two stacks S1 and S2 are created with the following contents: * S1: Rome, London * S2: Lisbon, Paris * as items are inserted in a LIFO/FILO manner * the item Lisbon is then retrieved and output/returned (S2 contents remain unchanged) |

**(b)** Three items in a list, L, are to be output in the order: first item, third item, second item.

Write an algorithm in pseudocode, using the operations given above, that would use a   
 stack R to carry this out. You may use L1, L2 and L3 to refer to the items in the list. [4]

|  |
| --- |
| Create(R)  Insert(R, L2)  Insert(R, L3)  Insert(R, L1)  WHILE NOT EmptyStack(R)  Retrieve(R)  Delete(R, 1) // assume function remove topmost/first item from List R |

**(c)** Explain why a list ADT is suitable for implementing a stack ADT. [3]

|  |
| --- |
| * stack ADT can be implemented using a list ADT by constraining the insertion and deletion of items **from one end** (front/top) only * eg Insert(S, item) will implement Push(S) while Delete(S, 1) will implement Pop(S) * the implementation is efficient as only pointers/references need to be adjusted after insertion/deletion and items need not be physically shifted |

**(d)** A programmer decides to implement the stack ADT by writing program code and   
 including the use of arrays.

**(i)** Explain two possible problems in using arrays to implement a stack ADT. [5]

|  |
| --- |
| * **static allocation** data structure - memory requirements eg stack size needs to be **predetermined** at program design/development time * hence does not allow for growth/expansion or result in inefficient use of space if large allocated amount is not used * have to reserve large contiguous amount of memory as items are accessed by array index/subscript in **constant time** * **expensive reorganization** of stack items as insertion and deletion at one end (top) results in shifting of all items up and down by one * leads to **inefficient** insert and delete running times |

**(ii)** Write an algorithm in pseudocode for the Insert operation of this implementation. [3]

|  |
| --- |
| Insert(S, item)  // make space for pending insertion, assume 0-based index, top = 0  for i = 1 to Size(S) // assume there is a function Size() which gives number of items in S  S[i] = S[i-1]  S[0] = item |

**(iii)** Program code should be written to display clarity. Describe three features of the   
 final program code for this implementation that would achieve this goal. [6]

|  |
| --- |
| * meaningful identifier names * eg * appropriate use of whitespace (indentation, blank lines) * eg * appropriate annotation/comments * eg |

**2010P1**

**6** A hash table has an index range of 1 to 400. The following pseudocode describes an   
 algorithm for searching the table using a hashing method. It is assumed that the key is   
 present in the table.

1. index = hash(key)

2. while table(index,1) <> key

3.  index = index + 1

4. endwhile

5. value = table(index,2)

**(a)** Explain the purpose of:

**(i)** line 1

|  |
| --- |
| Obtains the address of the desired target by passing the key through a hashing function |

**(ii)** line 2

|  |
| --- |
| To check if the key matches the key in the address found, if not, the program will loop until the key matches |

**(iii)** line 3

|  |
| --- |
| In the case of a collision where the key is does not match the one in the address, linear probing is carried out and the index is incremented by one until the key matches. |

**(iv)** line 5

|  |
| --- |
| Assigns to value the record stored at the correct address with matching key to be returned |

  in this algorithm. [8]

**(b)** The algorithm fails to handle the upper limit on the range of the index. What   
 modification to the algorithm is required to overcome this problem? [3]

|  |
| --- |
| Add another while loop after line 3 but before line 4:  if index > 400:  index = 1 |

**(c)** Explain how a new item can be added to a hash table. [4]

|  |
| --- |
|  |

**(d)** Describe **two** examples of the use of hash tables. [4]

|  |
| --- |
| Storing of information of library books with the index based on hashing the ISBN. Retrieval of information is required to be fast, which hash tables fulfils (with efficiency of O(1) if no collisions)  Storing of information of stock in an e-commerce business with the index based on item code. |

**2010P2**

An apartment block in the centre of a city consists of a large number of departments. There is also a pool with fitness centre in the basement for the use of residents.

The complex is administered from an office by a small group of employees.

**1** Each of the residents of the apartments has their information stored in a file. This   
 includes the start date of their residency.

**(a)** When data is entered into the file it must be validated and verified.

**(i)** Explain the difference between validation and verification of data. [2]

|  |
| --- |
| Data validation is the automated checking of data to ensure data values are **acceptable /reasonable**.  It ensures data is acceptable, but does not indicate whether the data actually exists.  Data verification is the process of **ensuring transferred data matches source data**, which is the checking of whether start data lies within actual range of given dates.  For example, if a new resident is registering, data validation would be the checking of whether the contact numbers are of numeric data types, hence whether they are valid, while data verification would be to check the provided NRIC of the resident with the NRIC on his/her identification card (manual data verification in this case)..  Data validation is automated (processed by the computer), while data verification can be both automated (checking transferred checksum and destination checksum) or manual (double data entry). |

**(ii)** Describe **two** validation tests that can be performed on the start date. [4]

|  |
| --- |
| range check:  check if the input date lies within the defined range of the possible start dates.  ie. range would be from the release of the apartment to the current date.  Assuming that the start date is DD/MM/YYYY, expression check can be performed:  check whether provided date is of correct composition; characters in index 0-1,3-4,6-9 are numeric, while characters in index 2,5 are ‘/’. |

**(iii)** Describe how a monthly payment from a resident can be verified when it is input   
 to the system. [2]

|  |
| --- |
|  |

**(b) (i)** Describe the difference between the use of fixed and variable length records with   
 reference to the file. [4]

|  |
| --- |
|  |

**(ii)** The records are currently stored in order of the apartment numbers, but they are   
 usually accessed according to the name of the resident.

Explain the difference between the file access which uses the resident name to   
 search for a record and file access which uses the apartment number. [2]

|  |
| --- |
| By searching using resident name, records are linearly traversed as the records are not ordered by resident name. Thus, an average of n/2 records need to be accessed to search for the record by resident name.  Searching by apartment number is faster, with records guaranteed to be found within O(log2(n)). The record may be found using binary search since the data is ordered by apartment number. The size of the problem is split into two every time a search is conducted, which makes the search more efficient. |

**2010P2**

**2** A decision has been made to change the method of file storage and access.

The records in the file are to be stored into alphabetical order of the resident's name.

**(a)** Using the following list of names as an example, show how the records can be sorted   
 into alphabetical order using an insertion sort.

GRA, CHR, DAV, SAR, TOM, KAT

|  |
| --- |
|  |

**(b)** Residents sometimes make requests for maintenance on their apartments. Each   
 request is given a priority number ranging from 1, for failure of the air conditioning, to   
 10, for a dripping tap. Each request is stored in a linked list in order of priorities. Jobs   
 with equal priority are stored in order of the date that they have been submitted.

Describe an algorithm to insert a new job into the list. [6]

|  |
| --- |
|  |

**2010P2**

**3** When a resident wants to use the pool with fitness centre, they use a swipe card   
 together with a 4-digit PIN to gain access. Access is only allowed if there are fewer than   
 20 people already using the facilities, in order to avoid overcrowding. Access is also   
 restricted to one person per card. If maintenance is being carried out then access is   
 denied and a message is output to a screen asking the resident to return in 30 minutes.   
 In order to exit the facility the swipe card is used again.

**(a) (i)** Describe the input devices in this system and how they are used. [4]

|  |
| --- |
| card reader?  PIN pads? |

**(ii)** The data collected or stored by the system includes:

* the date
* the total amount of time spent in the facility in hours (for example 3.25 hours)
* whether or not a resident is currently in the facility

State the data types which would be used for each of these items of data, giving   
 reasons for your answers. [6]

|  |
| --- |
|  |

**(b)** Use appropriate variable names and pseudocode to describe an algorithm to control   
 the entry system to the pool with fitness centre. [8]

|  |
| --- |
|  |

**2010P2**

**4** Residents make their payments monthly. Payments must be made by the last day of the   
 month. Details of the payments are stored on a transaction file. The individual payments   
 are then used to update the payment file overnight at the end of the month.

**~~(a)~~****~~(i)~~** ~~State the type of operating system used for this task, justifying your answer. [3]~~

**(ii)** When the payment file is updated it is done by updating the payment file with the   
 transaction file (TF).

Describe how the payment file and the TF can be processed to produce an   
 updated payment file. [7]

|  |
| --- |
|  |

**(b)** Following the updating of the payment file it is necessary to contact all the residents   
 who have not yet paid.

Describe how the details of these residents can be extracted from the files. [5]

|  |
| --- |
|  |

**2009P1**

**2** An examination board decides that its examiners who have internet access will   
 enter examination marks on-line rather than using the existing manual methods for data   
 capture.

**(a)** Examiners who do not have access to the internet must enter their marks on a form.   
 State and justify two input methods that could be used. [4]

|  |
| --- |
|  |

**(b)** Explain the difference between data verification and data validation. [4]

|  |
| --- |
| data verification   * process to confirm if input data entered is same as source data * typically involve manual inspection (by human) * may be caused by poor quality of source documents or human errors eg transposition or transcription errors * performed after data is input to the system   data validation   * process to confirm if input data conforms to required specifications in presence, existence, accuracy, length, range, format, etc. * typically validated by computerized means (program testing) * mainly caused by human errors during data input * performed at time of data entry |

**(c) (i)** Describe a verification method which can be used when marks are entered by   
 those examiners with internet access. [3]

|  |
| --- |
| * examiner report: enter data twice by same person (efficient?) * generate checksums from both source and input data, possibly performed by different people at different locations * verify that checksum from source data corresponds with checksum from input data * small variances in data produce vastly different checksums |

**(ii)** Describe a verification method which can be used by examiners who do not have   
 internet access. [2]

|  |
| --- |
| * examiner report: manual checking of entered data on form with marked exam paper * get second person to re-enter or verify data (if examiners do not work in isolation from home) |

**(d)** State two validation checks that could be used in this application. For each check   
 explain why it is appropriate. [4]

|  |
| --- |
| * range check - marks are in the appropriate range e.g. 0 - 100, so that correct statistics (eg maximum, minimum, average) can be generated * data type check - all input is numeric as marks should not be non-numeric type, leading to arithmetic processing errors * presence check - input is not absent to check that there is no omission error due to carelessness |

**2009P1**

**3** A recursive algorithm for finding a value, SearchItem, in an ordered array, X, is as   
 follows:

   Search(Low, High)

   Mid = (Low + High) div 2

   If X(Mid) = SearchItem then output "found" : exit

   If X(Mid) > SearchItem then Search(Low, Mid-1)

 Else Search(Mid+1, High)

   End Search

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

   Using the above algorithm:

**(a)** Explain what is meant by a recursive algorithm. [1]

|  |
| --- |
| a recursive algorithm is one that   * calls itself with a smaller similar subproblem (recursive case(s) eg If X(Mid) > SearchItem then Seach(Low, Mid-1) will discard half of input and recurse into appropriate half * until it meets its terminating case(s) which can be solved directly eg If X(mid) = SearchItem them output “found” : exit - item is found and algorithm returns result * and backtracks to eventually arrive at solution to the original problem |

**(b)** Describe what might occur during execution with an incorrectly written recursive   
 routine. [3]

|  |
| --- |
| * program may return wrong result and exit normally (logic error) * base/terminating case not met leading to infinite loop, program never exits * until program may run out of system allocated resources i.e. out of memory error (too many copies of program state and variables pushed into runtime stack, causing stack overflow exception) * program may crash or make other applications and overall system performance sluggish or system crash |

   Array X has 15 elements and the subscripts start at 1.

**(c)** If the algorithm was used to search the array X for the value stored at X(3) state the   
 calls to Search as the recursion executes. [4]

|  |
| --- |
| Search(1,15), mid = 8, so call  Search(1,7), mid = 4, so call  Search(1,3), mid = 2, so call  Search(3,3), found, terminating case and exit |

**(d)** The algorithm does not handle the case where SearchItem is not present in X.   
 Indicate what changes need to be made to Search to rectify this problem. [3]

|  |
| --- |
| if low > high then output “not found” : exit  else // terminating case when found |

**(e)** For this method of searching state the maximum number of comparisons and the   
 minimum number of comparisons for array X, justifying your answers. [5]

|  |
| --- |
| maximum = log2 n + 1  when n is 15 (input size), maximum = 4 + 1 = 5  log2 n is depth/height of binary search tree, + 1 for additional comparison for item not in binary search tree  minimum = 1 // first element in the middle is target |

**2009P1**

**5** A program contains the following code:

   if x =< a + b / c then

   begin

 s = t + 2

   end

   u = u + 1

**(a)** By using the above code to illustrate your answer, describe the four types of   
 program error:

**(i)** syntax,

|  |
| --- |
| * should be <= instead of =< * error due to violation of rules/grammar of programming language * flagged by language translator to produce diagnostic error message indicating source and possible cause of error |

**(ii)** semantic,

|  |
| --- |
| * s = t + 2 could be a potential semantic error if s is character type and t is integer type * error in meaning of program statement eg incompatible data types * flagged by language translator |

**(iii)** logic,

|  |
| --- |
| * a + b / c should be (a + b) / c as multiplication/division takes precedence over addition/subtraction, so the meaning is misinterpreted, or * s = t + 2 could be actually s = t \* 2, due to wrong formulation, or * u = u + 1 should be inside the if condition * not flagged by language translator, program translates successfully but produces wrong result when run |

**(iv)** arithmetic.

[8]

|  |
| --- |
| * a + b / c division by zero error if c = 0, * not flagged by language translator, will result in runtime/exception error as result is undefined for special cases of input * other examples of arithmetic errors: overflow, underflow |

**(b)** Explain how a programmer can debug a syntax error. [2]

|  |
| --- |
| * language translator will generate diagnostic message when it detects a syntax error, the message often contains the location of the error (line number), the part of code which is in error, and a suggestion of the possible cause of error * programmer can make use of this info to locate the line number, (sometimes error occurs in the preceding line), rectify the error before translating the program again |

**(c)** How might a programmer use black box testing when developing a program? [4]

|  |
| --- |
| * provide possible input-output specifications and see if this matches with expected results generated by module * can be used after white box testing / unit testing is performed to test the interfaces/dependencies between modules (integration/system testing) * can be used as dummy stubs for modules that have not been developed so that other parts of the system can be tested * can be part of top-down modular development to provide high-level view of system |

**(d)** Describe two debugging techniques that a programmer could use and include details   
 of errors that might be detected. [4]

|  |
| --- |
| * tracing using print statements to output values of variables at suitable junctures of program (eg before/after loop and before/after function call / expression) - able to detect if variables contain correct values to meet criteria for subsequent conditional/loop controls * programmer can set breakpoints and use the function stack call trace to inspect if functions are called with the correct parameters - able to detect wrong function calls or parameter values * programmer can use the step-over and step-into features to determine line by line execution of code within the calling and called functions (step-over if called function code need not be traced, step-into if called function code needs to be traced) |

**(e)** Name and describe the two testing strategies that can be used immediately before a   
 major item of software is ready for sale. [5]

|  |
| --- |
| * beta testing * user acceptance testing |

**2009P1**

**7.** A programmer has to write a program for a large business. The program is to process   
 and store payroll data about the employees of the business.

**(a)** State **three** elementary data types the programmer will need to use in the program,   
 giving an example of the use of each. [6]

|  |
| --- |
| * integer - number of hours worked assuming that whole numbers are counted only * float/double/real - salary as currency figures typically involve fractional numbers * boolean - paid to determine if an employee has been paid (true or false) * char - employee type eg ‘H’ for hourly rate employee, ‘S’ for salaried employee |

**(b)** Explain what is meant by 'composite data type'. Give **two** examples of composite   
 data type that might be used in this program. [4]

|  |
| --- |
| * consists of one or more elementary data types grouped logically * may or may not be homogeneous (eg array vs record) * string eg name consists of a sequence of alphabetical characters * array (homogeneous) eg array of employees for collective processing in a loop * record (heterogeneous) eg employee record may contain employee id (integer), name (string), type (character), salary (float/double/real) |

**(c)** The programmer implements much of the code by using both procedures and

functions.

Two features to be included in the program are:

* calculating a total tax payable
* outputting a list of employees in alphabetical order

**(i)** Explain the difference between procedures and functions. [2]

|  |
| --- |
| * procedure - method/subroutine which does not return a value e.g. PrintMenu() output list of menu options * function - method/subroutine which returns a single value e.g. CalcArea() returns computed area |

**(ii)** For each of the two features described above state whether a procedure or   
 function would be used. Justify your answers. [4]

|  |
| --- |
| * calculating a total tax payable - function as amount of tax payable will be computed within the function and returned to the calling routine * outputting a list of employees in alphabetical order - procedure as the function can print the list and need not return a result to the calling routine |

**(d)** Parameters can be passed to a procedure by using pass-by-value or   
 pass-by-reference.

Explain the difference between these two methods. [4]

|  |
| --- |
| * pass by reference - array of employees * pass by value - hourly rate   justify (definition, characteristics, inferences, examples, pros & cons) |

**2009P2**

A publishing company specializes in publishing books about Singapore. It also has two shops through which its books are sold.

**1** The catalogue of books available at the shop is stored in a computer system as a   
 direct access file.

One index to this file is arranged as a binary tree of the authors' names according to   
 the following algorithm.

READ N

REPEAT

COMPARE TO ROOT NODE

IF N > ROOT NODE THEN FOLLOW RIGHT POINTER TO NEW ROOT NODE

ELSE FOLLOW LEFT POINTER TO NEW ROOT NODE

END IF

UNTIL ROOT NODE EMPTY

INSERT N

END

**(a)** The algorithm is not well presented. Identify **two** techniques which should have been   
 used to improve the understanding of the reader.

State why each of your chosen techniques is important.   [4]

|  |
| --- |
| * white space (indentation)   + make clear which segment belongs under which code block * use mixture of upper and lowercase   + easier to read due to font characteristics eg kerning * use pseudo-function call eg   ELSE FOLLOW LEFT POINTER TO NEW ROOT NODE  can be written as  Search(node.left) or goToNode(node.left)   * makes algorithm more concise and easier to understand |

**(b)** The algorithm assumes that authors' names are unique.

Describe how the algorithm could be altered to identify, as an error, when an author's   
 name is input which already exists in the tree. [3]

|  |
| --- |
| COMPARE TO ROOT NODE  IF N = ROOT NODE THEN OUTPUT ERROR “DUPLICATE AUTHOR NAME!”, EXIT  ELSE IF N > ROOT |

**(c)** Explain why the use of a REPEAT...UNTIL construct will fail if the tree is empty.

State another iteration structure which will overcome this problem. [3]

|  |
| --- |
| * DO ... WHILE (condition) is equivalent to REPEAT ... UNTIL (condition)   + loop body is performed at least once, since condition checking is done at the end   + thus cannot perform action or will generate error if tree is empty   + more appropriate to use WHILE (condition) which is performed 0 or more times, since condition checking is done before deciding to perform loop body |

**(d)** The following names

TAI, CHO, SIM, DAN

are to be inserted into a subtree, whose root node is MAI.

Using the algorithm, draw the completed subtree. [2]

|  |
| --- |
|  |

**(e)** Describe an algorithm for reading the complete set of names, stored in the tree, in   
 alphabetical order. (Note: only an outline description of the algorithm is required.)[3]

|  |
| --- |
| inorder traversal  Inorder(T)  if T != NULL // not leaf node, continue traversing inorderly  Inorder(T.left)  Process T.data  Inorder(T.right)  Algorithm will traverse left subtree recursively until leaf node, process data and traverse right subtree recursively until leaf node. This will generate an alphabetically sorted list of names. |

**2009P2**

**2** The company decides to shut one of the shops and to sell through the internet instead.

**(a)** Explain the effects of this decision on people who work for the company.     [4]

|  |
| --- |
| * unemployment - eliminate need for sales personnel manning the shop * retraining for remaining employees to learn about new online sales procedures * may need to recruit new employees with relevant new skill set (eg web design, database maintenance) * reshuffling/restructuring of organization structure to cater to new business mode * option to work from home / telecommuting |

The new system will require new software to be written. A systems analyst is   
 employed to manage the production of the software.

**(b)** Explain why a clear statement of the problem to be solved is essential.   [2]

|  |
| --- |
| * systems analyst to gather input from and meet requirements of client * client understand the possibilities, potential and limitations of the solution * both arrive at objectives and scope of the problem |

**(c)** The analyst must understand the data requirements of the solution.

* + A customer will be able to place an order either by telephone or via the internet
  + The order will be placed in a file to be dealt with by the warehouse staff
  + Acknowledgement of the order will be sent to the customer as an email
  + After completion of the order the customer details will be stored in a customer file

Determine the flow of data in the solution and draw a diagram to represent it.     [4]

|  |
| --- |
|  |

**(d)** The analyst decides to adopt a top-down approach to the design.

   Explain what is meant by a top-down approach. [2]

|  |
| --- |
| * break down large complex problem into smaller simpler subproblems * repeat until individual problems can be translated to a single solvable unit/module as an algorithm |

**(e)** A procedure is written to look up the customer record in the customer file.

The procedure then adds the value of the current order to the total ordered by   
 the customer this year.

This determines whether or not a discount is payable.

**(i)** Give an example of a variable which should be declared as a global variable,   
 justifying your answer. [2]

|  |
| --- |
| * total ordered by customer this year   + must be accessible by all modules   + exist throughout lifetime of program   + convenient to use rather than passed as parameter to every module |

**(ii)** Give an example of a parameter that should be passed by value, justifying your   
 answer. [2]

|  |
| --- |
| * current order   + value should not be changed outside module   + subroutine can work with its own local copy avoiding side effects |

**2009P2**

**3**  A file of books that are published by the company is kept.

Each book has a code number (C).

A two-dimensional array SALES(X,Y) is kept where X is the total number of different   
 books published.

SALES(X,1) stores the code number of the book and SALES(X,2) stores the   
 total number of that book sold.

Each month the array is ordered so that SALES(1,1) stores the code number of the   
 book with the largest sales.

**(a)** Produce an algorithm to sort the array into the required order using a bubble sort.    [6]

|  |
| --- |
| # e.g. SALES = [[id1, num1], [id2, num2], [id3, num3]]  def bubble\_sort(array):  sort = False  while not sort:  sort = True  for x in range(len(array)-1, 0, -1):  if array[x][1] > array[x-1][1]:  array[x], array[x-1] = array[x-1], array[x]  sort = False  if sort:  return array  e.g. SALES = [[1, 4], [2, 6], [3, 10], [4, 1]]  Output - [[3, 10], [2, 6], [1, 4], [4, 1]] |

**(b)** The algorithm is coded and the finished code for the sort must be tested.

The test data must test the purpose of the algorithm to possible values of X when the   
 array is created.

**(i)** State **three** different sets of test data that can be used. Justify each of your   
 choices. [6]

|  |
| --- |
| * best case - already sorted, no swap needs to be made, complete in 1 pass, linear time efficency * worst case - reverse sorted, to make maximum number of swaps, complete in n passes, quadratic time efficiency * normal/random case - partially sorted, efficiency lies between linear time and quadratic time |

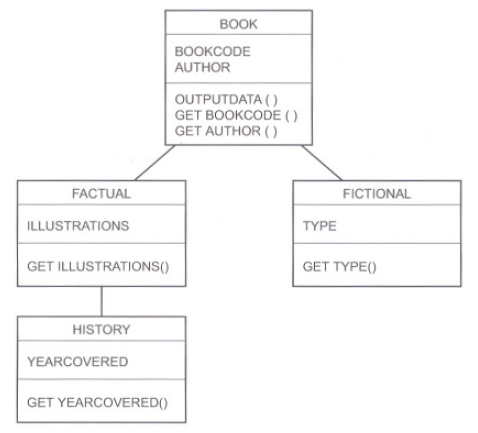
**(ii)** The code for the bubble sort has been fully tested and seems to work   
 perfectly. However, when it is combined with other modules to form the complete   
 program it fails.

Explain why this has happened. [2]

|  |
| --- |
|  |

**2009P2**

**4**  Some of the information about books is manipulated using classes. The following   
 diagram shows the relationship between some of the classes.



**(a)** Describe and define another subclass that can be included on the diagram and state   
 where it fits.     [4]

|  |
| --- |
| SCIENCE FICTION as a subclass of FICTIONAL   * data - Technology/Cliche Used * methods - get\_Technology/Cliche Used, set\_Technology/Cliche Used |

**(b)** With reference to the diagram explain the terms:

**(i)** data encapsulation

|  |
| --- |
|  |

**(ii)** inheritance     [4]

|  |
| --- |
|  |

**2009P2**

**7 (a)** The file that stores details of all the company's publications is organized as a   
 sequential file. This file is ordered on the book's 6-digit identification number.

Describe an algorithm for adding a new book to the file.  [6]

|  |
| --- |
| open book file  read first record and compare id with new book’s id  loop while current record id < new book id  write record to new sequential file  if current record id = new book id (duplicate record)  error  else  write new book record to new book file  loop while not end of file book file  write remaining records to new book file |

**(b)** All orders from customers are placed on the order file.

**(i)** Describe how the order file is arranged as a random access file using hashing. [4]

|  |
| --- |
| * devise a suitable hashing algorithm eg using modulo arithmetic mod n, where n is a suitably large prime number to minimize collision * apply hashing function to record key to determine physical record address in prime area * if location is free, then store record * if location is occupied (i.e. collision), go to overflow area and store record as a sequential list |

**(ii)** Explain why the file is designed to be random access rather than sequential   
 access. [2]

|  |
| --- |
| * allows for efficient one-time lookup of customer’s order by direct computation of record address by applying hashing function to record key * access time is independent of record physical location |

**(c)** The customer, order and book files are part of a database. Copy and complete   
 the diagram to show the relationships between the customer and order files and the   
 order and book files.   [4]



|  |
| --- |
|  |

**2008P1**

**2** An examination system requires that each candidate has a unique ID consisting of six   
 digits.

   The first two digits are the school code and the next three digits are the candidate's   
 unique number within the school. The sixth digit is a check digit. The validity of the   
 check digit is computed by the following process:

* multiplying each digit by its position in the number (counting from the right);
* summing the results of the multiplications;
* calculating the sum mod 11;
* if the final result is zero then the ID is valid.

**(a)** Show whether the following IDs are valid in this system.

**(i)** 280410

|  |
| --- |
| 280410 - valid as final result = 0 |

**(ii)** 503313

[4]

|  |
| --- |
| 503313 - invalid as final result != 0 |

**(b)** Describe, with examples, two data entry errors that a check digit helps to detect.   [4]

|  |
| --- |
| * transposition error eg 280410 is entered as 280140, final result != 0 therefore invalid * transcription error eg mistake 8 for 3 eg 280410 is entered as 230410 |

**(c)** Write an algorithm, in any appropriate form, that will calculate the correct check digit   
 for a new candidate.                                                                                               [6]

|  |
| --- |
| input/get school code and candidate unique number  multiply each digit by the weights 6, 5, 4, 3, 2  compute sum of products  get remainder when sum of products is divided by 11  check digit = 11 - remainder  12345  65432  6 + 10 + 12 + 12 + 10  50 mod 11  6  11 - 6  123455 |

**(d)** The algorithm produced in (c) is to be implemented in a program as a module.

**(i)** Justify the most appropriate type of module to use.                       [2]

|  |
| --- |
| * function - returns check digit as output of module |

**(ii)** State the parameter for this module.                                                               [1]

|  |
| --- |
| * first 5 digits of id i.e. 2-digit school code concatenated with 3-digit candidate number |

**(e)** Give an appropriate data type for the ID and justify your choice.     [2]

|  |
| --- |
| * integer - arithmetic operation will be performed on parameter * in practice, this approach may be more complicated due to the need to calculate the positional value of the digits within the id code * alternative: string - use substring function to extract positional value and cast to integer and multiply by appropriate weight |

**2008P1**

**4** A bank has 200,000 customers. Each customer has a unique ID.

Each time a customer uses an ATM to obtain cash their current balance must be   
 checked.

The balance is stored in a file that has the customer ID and PIN as additional fields.

**(a)** Explain why a random organization would be appropriate for this file used in the   
 ATM application.     [3]

|  |
| --- |
| * customer query transaction is a one-time lookup activity which should produce reasonable response time independent of physical record location * random file organization with hashing function which maps record key to physical address possesses this characteristic * access time is fast as it is a mathematical algorithm which can be directly computed |

**(b)** Describe an algorithm that will locate a record in the random file.             [5]

|  |
| --- |
| * apply hashing function to record key to get physical address * if prime area does not contain any record, return not found * else if prime area contains record with record key, retrieve record from secondary storage to main memory for processing * else if prime area contains record but of different record key (i.e. collision), access overflow area to linearly search for required record (assuming collided records are stored in overflow area in a sequential list) * if overflow area does not contain record, return not found * else retrieve record from secondary storage to main memory for processing |

**(c)** Why would the random file need to be recreated periodically and how would this be   
 done? [4]

|  |
| --- |
| * over time as new records are added, there may be many collisions leading to slower access times if records have to be often retrieved from the overflow area * may also have to provide buffer for future record growth * need to reorganize records such that fewer collisions result leading to faster access times * one way is to devise a better hashing function by studying the characteristics of the data * eg mod n where n is a suitably large prime number which will minimize collisions * another way could be to design another collision resolution strategy e.g. each prime area record location may have its own overflow list |

   Details of withdrawals are held in a customer master file.

**(d)** Explain why the records in the customer master file are variable length.     [2]

|  |
| --- |
| * fields like customer name and customer address occupy different lengths * it will be more efficient to store these as variable length records to save space |

**(e)** Describe how the computer system may determine where one variable length record   
 ends and another begins.                                                                                         [2]

|  |
| --- |
| * variable length records can be stored in a delimited file eg csv (comma separated variables) in which fields are separated by a delimiter eg , or ; or | * once the correct number of delimiters are read, then the computer would know that it has finished reading the current record and move on to the next record |

**2008P1**

**5** A programmer is going to write an application, using an object-oriented programming   
 language, to store details of a large collection of different types of DVDs, including music   
 DVDs and film DVDs.

**(a)** Explain, using an example from the application in each case, what is meant by:

**(i)** class;                                                                                                    [3]

|  |
| --- |
| * blueprint/template or blueprints for creating objects of the same type * provides encapsulation to group data (state) and methods (behaviour) within an entity * eg DVD is a class which will contain data (eg. ) and methods (eg. ) * enforces principle of data/information hiding by restricting access of private data to public methods (interface) |

**(ii)** object;                                                                                                    [3]

|  |
| --- |
| * runtime instantiation of a class * represents the entity that is manipulated in the program * MusicDVD mus\_dvd1 will instantiate an object mus\_dvd1 of the class DVD which will contain data (eg. ) and methods (eg. ) |

**(iii)** inheritance;                                                                                             [3]

|  |
| --- |
|  |

**(iv)** polymorphism.                                                                                       [3]

|  |
| --- |
|  |

**(b)** Show, using a diagram, a method of specifying the class: film DVD.                       [4]

|  |
| --- |
| * UML class diagram * show the 3 sections: class name, private data, public methods * appropriate data and methods (inherited and new) * show inheritance (upward pointing arrow to superclass DVD)   film DVD  private data - Title, Director, RunningTime  public methods - constructor, get\_title/director/running\_time(), set\_title/director/running\_time() |

**(c)** Give two advantages, to the programmer, of using an object-oriented   
 programming language.       [2]

|  |
| --- |
| * more natural/direct mapping of classes/objects to entities in problem domain * sandbox security protection (encapsulation) so that private data are protected from unauthorized access/modification and only accessible via public methods * inheritance allows code reuse eg common data and methods in superclass * polymorphism allows concise and powerful mechanism to invoke different classes’ methods of the same name * allows concurrent and distributed control/processing * event-driven model by message passing |

**2008P1**

**7** An Abstract Data Type (ADT) consists of both data type and associated operations.

   A linked list ADT has the following operations defined:

(i) Create(L) - creates an empty linked list L;

(ii) Insert(L, item, P) - insert new value, Item, into linked list L so that it is at position P in   
 the linked list;

(iii) Retrieve(L, P) - get a copy of the item at position P in the linked list L;

(iv) IsEmptyList(L) - returns true if linked list L is empty.

**(a)** Describe the outcome after the following operations have been carried out:

   Create(ListOne)

   Create(ListTwo)

   Insert(ListOne, Fred, 1)

   Insert(ListTwo, Fred, 1)

   Insert(ListOne, Jim, 2)

   Insert(ListOne, Alan, 1)

   Retrieve(ListOne, 2)                                                                                           [5]

|  |
| --- |
| ListOne - Alan, Fred, Jim  ListTwo - Fred  Retrieve(ListOne, 2) - Fred |

**(b)** Write pseudocode, using the operations given above, that would copy the items in   
 ListOne and place them in ListTwo in reverse order.                                               [6]

|  |
| --- |
| endOfList = False  i = 1  while not endOfList  item = Retrieve(ListOne, i)  if item != NULL  Insert(ListTwo, item, 1)  i += 1  else:  endOfList = True |

**(c)** A stack ADT has a Create operation. Give three other operations that would be   
 included in a stack ADT.                                                                       [3]

|  |
| --- |
| Push(Stack, item) // insert item into stack  Pop(Stack) // delete item from stack  Peek(Stack) // retrieve (but do not delete) top item from stack |

**(d)** For two of the operations that you have given in (c) show how they could be   
 implemented using the linked list ADT operations.                                                   [4]

|  |
| --- |
| Push(Stack, item)  Insert(Stack, item, 1)  Pop(Stack)  if isEmptyList(Stack)  !!! no delete  Peek(Stack)  if isEmptyList(Stack)  return error // cannot peek from empty stack  else  Retrieve(Stack, 1) |

**(e)** Give one advantage and one disadvantage of using an array data structure   
 when implementing the linked list ADT.                                                                   [2]

|  |
| --- |
| advantage   * fast/direct/random access using array subscript/index (constant access time) as items are stored in contiguous memory locations   disadvantage   * static/fixed memory allocation at compile time (program development), cannot cater to future growth * need large enough contiguous memory block to store array elements |

**2008P2**

**2.** After a customer has input the necessary information to the company website, the   
 system calculates the cost according to the following rules:

1. $5 for each game for 1 or 2 people;

2. an extra $2 for each person over 2 people;

3. a 10% reduction if more than one game is booked;

4. an additional 10% if the customer is a member.

The number of people is stored in N.

The number of games is stored in G.

Whether or nor the customer is a member is stored in J.

The total cost of the booking is stored in TOTAL.

**(a)** Explain why TOTAL is a more sensible variable name than N, G or J.                 [3]

|  |
| --- |
| * more indicative of purpose/meaning of variable, more descriptive identifier name which can communicate better the purpose of the variable to the programmer developing/ maintaining the program * facilitates readability by making code self-documenting * leading to improved maintainability |

**(b)** For each of the following variables state a suitable elementary data type, justifying   
 each of your choices.

**(i)** N

|  |
| --- |
| * integer as number of people must be a whole number |

**(ii)** J

|  |
| --- |
| * boolean as only two possible values depending on whether a customer is a member (TRUE) or not (FALSE) |

**(iii)** TOTAL

[6]

|  |
| --- |
| * float/double: the total cost of booking will have a fractional component as calculations relating to percentages involve division operations |

**(c)** The following pseudo code has been produced to calculate TOTAL.

The line numbers are to help with your answers.

 10 INPUT N, G

 20 PART\_TOTAL = 5 \* G

 30     IF N < 2 THEN ADDITIONAL = 2 \* N

 40     END IF

 50 PART\_TOTAL = TOTAL + ADDITIONAL

 60     IF G > 1 THEN PART\_TOTAL = .1 \* PART\_TOTAL

 70         IF J = TRUE THEN PART\_TOTAL = .1 \* PART\_TOTAL

 80         END IF

 90     END IF

100 TOTAL = PART\_TOTAL

110 OUTPUT TOTAL

120 END

**(i)** Explain what is happening in line 70.                                                       [2]

|  |
| --- |
| * line 70 - if customer is a member, then compute additional 10% discount amount |

**(ii)** There are two errors in line 30.

Rewrite the line as it should have been written.                                                 [2]

|  |
| --- |
| * line 30 - IF N > 2 THEN ADDITIONAL = 2 \* (N - 2) * explain - an extra $2 for each person over 2 people * original code wrongly specifies less than 2 people and multiply extra over all N people |

**(iii)** Lines 50 to 90 contain a number of errors.

Identify each error and state how it should be corrected so that the algorithm will   
 output the correct TOTAL.                                                                                 [6]

|  |
| --- |
| PART\_TOTAL = PART\_TOTAL + ADDITIONAL  IF G > 1 THEN PART\_TOTAL = 0.9 \* PART\_TOTAL  IF J = TRUE THEN PART\_TOTAL = 0.9 \* PART\_TOTAL  ENDIF  ENDIF   * TOTAL in line 50 does not exist, change to PART\_TOTAL * line 60 - 90: Since 0.1 \* PART\_TOTAL will reduce PART\_TOTAL to 10% of the original value while 0.9 \* PART\_TOTAL will reduce PART\_TOTAL to 90% of the original value which is a 10% reduction of PART\_TOTAL * de-indent line 60 - 90 (indentation not meaningful) |

**2008P2**

When a member inputs his membership number the system finds his record in order to use and amend some of the data. The system uses specific program modules to do this.

**4 (a)** State what is meant by the following:

**(i)** a record                                                                                                     [1]

|  |
| --- |
| * a record is composite data structure consisting of two or more related fields usually of different/heterogeneous data types * eg employee record contain employee id (integer), name (string), gender (character), dob (date) |

**(ii)** a program module                                                                                   [1]

|  |
| --- |
| * a program module is a subprogram/subroutine which takes in well-defined input parameters, perform processing on this data and produces results as output * eg calc\_area() module of a circle will accept radius as input, compute area and this as the result |

~~During the run of the search module the value in the program counter (PC) changes   
 from 400 to 295 during a single processor cycle.~~

**~~(b)~~** ~~If the PC begins the cycle set to 400, explain:~~

**~~(i)~~** ~~what the value 400 represents;                                      [2]~~

**~~(ii)~~** ~~what value you would expect it to be set to at the end of a normal cycle; [2]~~

**~~(c)~~** ~~State what is meant by the term 'processor cycle time' and why it is important when   
 selecting hardware for a task.                                                                             [2]~~

All the modules were thoroughly tested before being used in the system. Some of the   
 errors which were found were syntax and logic errors.

**(d)** By using an example of each, explain what is meant by:

**(i)** a syntax error;

|  |
| --- |
| * error due to violating rules/grammar of a programming language * eg missing statement terminators such as semi-colon (Java, C++), illegal identifier names (variable starting with numbers 2num, containing illegal characters 5@sum, unmatched parentheses/braces * language translator (compiler/interpreter) stop translation process and produce diagnostic error messages * to indicate source of error (line number) and possible cause of error |

**(ii)** a logic error.

[4]

|  |
| --- |
| * program translates successfully but produces wrong results when executed * eg area\_circle = pi \* r \* r is written as area\_circle = pi + r + r * usually due to unintentional wrong formulation of expression * can only be resolved by human inspection and testing |

**~~(e)~~** ~~The language used to write the modules defines a variable name as~~

* + ~~any number of LETTERs (not zero) followed by~~
  + ~~0, 1 or 2 DIGITs followed by~~
  + ~~a % symbol OR a # symbol OR nothing.~~

~~Draw a syntax diagram, using LETTER, DIGIT, % and # to define a variable name. [6]~~

Each member of the club is placed in one of the four leagues.

The members in each league are arranged in order of the average number of points they score per game.

**2008P2**

**5** The membership number and average score are stored as elements in a linked list with   
 the highest average score first in the list.

Every time a member submits a new score their new average is calculated.

After the average score has been re-calculated it is necessary to adjust the linked list   
 to keep it accurate.

Explain the stages necessary to produce a new, updated, linked list.

The answer should be an explanation rather than a detailed algorithm.                       [8]

|  |
| --- |
| * sequential search to node containing membership number with new score (since list is not sorted by membership number), keep track of this location/position, call this node * compute and store new average in node * compare new average with start of list (first node)’s highest average * traverse until node with average just before the new average, call this curr * set node.link to point to curr.link * set curr.link to point to node |

**2008P2**

**7** At end end of each season the four leagues are merged into one, in order of player   
 average, with the highest average first. The members are then divided into 4 new   
 leagues, with the first quarter of members being in league 1, the second being in league   
 2 and so on.

Using A, B, C, D as the names of the original four leagues and 1, 2, 3, 4 as the names   
 of the new leagues, write an algorithm which will create the new leagues according to   
 the rules given.

Your algorithm may be presented in whatever form you wish.                                   [10]

|  |
| --- |
| one way   * store each league’s info in its own array A, B, C, D * create large array X with length = combined array lengths of A, B, C, D * store contents of A, B, C, D into X in turn * sort X * store each quarter of X into new arrays 1, 2, 3, 4 |

**2007P1**

**3** The following items are to be placed on a binary tree structure: tian, ben, ming, wei, ang   
 and van.

**(a)** Draw the alphabetically ordered binary tree after all the items have been added. [3]

|  |
| --- |
|  |

**(b)** Write an algorithm for the insertion of an item into an alphabetically ordered binary   
 tree. [5]

|  |
| --- |
| node - data, left, right  Insert(T, item)  if T = NULL // leaf node, so insert  new node n  n.data = item  n.left = NULL  n.right = NULL  else  if item < T.data // go to left subtree  Insert(T.left, item)  else // go to right subtree  Insert(T.right, item) |

**(c)** Write down the output if the tree is traversed using:

**(i)** In-order

|  |
| --- |
| inorder - process left, node, right |

**(ii)** Pre-order

|  |
| --- |
| preorder - process node, left, right |

**(iii)** Post-order

      [6]

|  |
| --- |
| postorder - process left, right, node |

A binary tree abstract data type (ADT) is to be implemented by using three   
 one-dimensional arrays.

**(d)** Show how the above binary tree would be implemented using these three arrays. [6]

|  |
| --- |
| data array to store actual values  left array to store left children  right array to store right children  data[i], left[i], right[i] denote the date, link to left subtree and link to right subtree respectively  use -1 to denote NULL i.e. no child |

**(e)** State two operations that would be required for the ADT.                           [2]

|  |
| --- |
| Insert(t, v) // insert node with data v into binary tree t  Delete(t, v) // delete node with data v from binary tree t  Search(t, v) // search for node with data v in binary tree t |

**2007P1**

**5.** Fig. 5.1 shows three classes and some of their private attributes and public methods.

**(a)** Use this example to explain the following terms:

**(i)** encapsulation,

|  |
| --- |
| * bundling/grouping of data and methods to form a class / object type * Employee class has both data (Employee Name) and methods (eg Set\_Employee Name) * data/information hiding - private data accessible only via public methods (interface) * private Employee Name is only accessible via public methods Set\_Employee Name and Get\_Employee Name |

**(ii)** inheritance,

|  |
| --- |
| * subclass adopts/inherits all data and methods from superclass * subclass Payroll Employee will inherit Employee Name and all methods (eg Set\_Employee Name) from superclass Employee * subclass can also define its own data and methods * Payroll Employee will have rate\_of\_pay data and Get\_hourly\_rate\_of\_pay, Set\_hourly\_rate\_of\_pay and Show methods which are specific to itself |

**(iii)** polymorphism.

[6]

|  |
| --- |
| * with inheritance, ability of different (sub)classes to implement and invoke different methods using the same name * Payroll Employee and Salaried Employee will implement and invoke their own Show method * dynamic binding - behavior occurs at runtime when class is determined * eg emp.Show will determine which class emp belongs to during program execution |

**(b)** Explain, using the above example, the relationship between a class and an object. [3]

|  |
| --- |
| * a class is a blueprint/template to create objects * an object is an instance of a class * Employee emp - emp is an object of the class Employee, it has its own state (data value Employee Name eg “Peter”) and common behavior (methods eg Get\_Employee Name will return name “Peter”) |

**(c)** How is data hiding achieved in a class?                                                                 [2]

|  |
| --- |
| * use of the private scope specifier for data eg Employee Name * ensure private data is accessible only via public methods eg Set\_Employee Name and Get\_Employee Name * protects data from unauthorized access and accidental modification |

**(d)** Some employees who earn a salary can receive a yearly bonus which is some fixed   
 percentage of their salary. Describe what changes should be made to the above   
 diagram to accommodate this possibility.                                                                [3]

|  |
| --- |
| Changes to SalariedEmployee class:   * add private attribute Yearly Bonus * add public methods Get\_Yearly Bonus and Set\_Yearly Bonus |

**(e) (i)** Explain what is meant by method overloading.                                             [2]

|  |
| --- |
| * same method name with different method signature   + different number of parameters   + different data types of parameters   + different order of parameters   + different method return type |

**(ii)** Give an example of method overloading that would be appropriate for this   
 application. [2]

|  |
| --- |
| * multiple constructors (note: programming language specific eg C++, Java)   + default constructor Employee() if we do not know employee details at object creation time   + overloaded constructor Employee(EmployeeName) if employee name is provided at object creation time |

**2007P1**

**6** An array X contains N integers stored in ascending order. The following pseudocode   
 describes a function for finding the position, in X, of an integer whose value is stored in a   
 variable called item.

 first = 1

 last = N

 while first <= last

   mid = int((first + last)/2)

   if X(mid) = item then

     return mid

   if X(mid) < item then

     first = mid + 1

   else

     last = mid - 1

   endif

 endwhile

 return not found

**(a)** if N = 200 then state the elements of array X that are examined when searching for   
 an integer which is in the 87th element.                                                                   [4]

|  |
| --- |
| * 100th, 50th, 75th, 87th |

**(b)** If the search is for an integer which is not present, then explain how many elements   
 have to be examined before 'not found' is returned.                                       [2]

|  |
| --- |
| * log2 n + 1 (n = 200, 8 + 1 = 9 elements) * maximum 8 comparisons to search to leaf node (still found), 200 is between 27 and 28, +1 for 'not found' * or height of binary tree is 8 (round to nearest integer log2 200) which is the number of comparisons needed to reach the leaf node, +1 for 'not found' |

**(c)** State the difference between global and local variables. Give an example of a local   
 variable in this function.                                                                                     [3]

|  |
| --- |
| global variable   * exists throughout lifetime of program * used across multiple functions/modules * convenient to declare as global to avoid passing as parameter to every function/module * saves storage space for large commonly accessed data structures   local variable   * exists only within the program block in which it is declared * memory will be released upon exit of program block * used for intermediate processing within program block * will not be accidentally accessed/modified by other parts of program   eg mid is a local variable as it is declared and only accessible/used within this program block |

**(d)** For this function state two parameters that would be used and how they would be   
 passed.        [4]

|  |
| --- |
| N, number of items in array passed by value   * used to determine upper bound * original value not meant to be changes   item passed by value   * a copy of variable is made in function for intermediate processing * original value is not changed   array X passed by reference   * saves storage as array can be large * original array will be changed as there is a need for function to change it |

**2007P2**

**2** The refreshments are sold from stalls in the central area. Network terminals are used by   
 the assistants who work at each stall. Details of items bought are input at the terminal   
 which is then used to complete the transaction.

**(a)** Two input devices that could be used in this application are:

**(i)** bar code reader,

**(ii)** touch screen.

Describe these two input methods and justify the choice of touch screen input.      [6]

|  |
| --- |
| * bar code reader   + scanner that reads vertical parallel stripes of lines containing item identification information   + well established universal standards, formats and convention   + barcode is cheap and easy to generate/print using freely available software   + uses infrared technology and hence limited effective range and requires line of sight for reading   + barcode cannot be read reliably if tainted with dirt marks or has faded * touch screen   + input technology that utilizes human/stylus point and touch action for data entry   + user-friendly as there are visual cues/icons/graphics to indicate possible choices   + fixed number of choices reducing data entry errors and eliminating need for validation   + easier to maintain and more resistant to dirt/marks   touch screen is preferred in the refreshments stall context as   * more intuitive and no training required * good for fixed number of choices as can be programmed within interface * more resistant to dirt/mark/fluid * not feasible/cost-effective to label all refreshment items with barcode * save materials cost and time to generate and print barcode and manpower costs and time to label items |

**~~(b)~~** ~~The data collected by the terminal at the refreshment stall is stored for later   
 transmission and processing at a central location.~~

~~When a customer buys a ticket at one of the booking desks the data is sent   
 immediately for processing.~~

~~With reference to these examples, explain the need for and describe the main   
 characteristics of a:~~

**~~(i)~~** ~~batch operating system,   [2]~~

**~~(ii)~~** ~~multi-user operating system,                                                                              [2]~~

**~~(iii)~~** ~~distributed operating system.                                                                            [2]~~

**(c)** Data stored about items of refreshments include:

* Quantity (in kgs, or litres)
* Price
* Sell-by date
* Number in stock

**(i)** State the suitable data types for each of these items of data.                               [4]

|  |
| --- |
| * Quantity - float/double/real can be partial kgs/litres eg 1.5 kg/l * Price - float/double/real correct to 2 decimal places for currency amounts * Sell-by date - string (for programming languages without date routine/library, needs to be cast into suitable type eg integer for processing) or date (for those that support common date/time functions) * Number in stock - integer since generally dealing with whole numbers |

**(ii)** The name of the item is stored as a set of ASCII characters. Explain what is meant   
 by an ASCII character.                                                                                         [2]

|  |
| --- |
| * American Standard Code for Information Interchange * 7-bit encoding based on English alphabet * only allows 27 = 128 possible characters to be represented * since word size is 8 bits, last bit is added as check/parity bit * represents both printable (eg alphanumeric) and non-printable (eg delete, backspace) characters |

**2007P2**

**3** The details of items sold in the refreshments stalls are stored in a sequential file. Items   
 sold are identified by an identification code. Identification codes are created according   
 to the following rules :

* A code letter identifying the stall at which it is sold (W for items sold at stall number 1, Y for items sold at stall number 2, and Z if it is sold at both stalls);
* A code letter identifying the type of product ( A for sweets, B for drinks and C for all other items);
* A two digit code between 10 and 99 to distinguish between items of the same type.

So, YB23 would be a drink of flavour 23 ( perhaps orange) sold only at stall 2.

**~~(a)~~** ~~Given that~~

~~<NON ZERO DIGIT> ::= 1|2|3|4|5|6|7|8|9~~

~~<DIGIT> ::= 0|<NON ZERO DIGIT>~~

~~Create a set of rules in Backus-Naur Form (BNF) to define an identification code for   
 an item.      [5~~]

**(b)** Describe two validation checks which should be carried out on the identification code   
 when it is input to the system.                                                                                 [4]

|  |
| --- |
| * length check - check if identification code is exactly 4 valid characters * format/range check - check if code letter is only W, Y or Z; check if code letter is only A, B or C, etc. * presence check - if some input is entered * case check - if both uppercase and lowercase are accepted |

**(c)** Details of items are stored in a table, each item being identified by its code. The table   
 is arranged sequentially according to item identification codes.

Write an algorithm in pseudocode which will take , as input, the identification code of   
 an item and search the table sequentially to find details of the item.                       [6]

|  |
| --- |
| SequentialSearch(table,id)  position = 1  while (id of position in table != id) and (position < length of table)  increment position by 1  if item found  print details  else  print item cannot be found |

**(d)** The table is stored as a linked list using pointers from one item to the next.

Explain how a new item can be inserted into the correct place in the list.               [5]

|  |
| --- |
|  |

**2007P2**

**4** Customers who attend the complex regularly are given a loyalty card.

   This card is used to keep a record of visits made to the theatre and customers are   
 awarded points according to how much they spend. These points can be exchanged for   
 items from the stalls or for tickets to watch films.

**(a)** Each card is identified by a six digit number.

* The first digit is a 1 or a 2 signifying whether the cardholder is male or female.
* The second and third digits give the age of the cardholder.
* The remaining three digits are used to ensure that each card is unique.

So, 218031 belongs to an eighteen year old female.

When a card is used it is swiped through a card reader and the holder’s record is found.   
 The value of the purchase is then added to their record.

**(i)** Explain why a random file is used to store the records as opposed to a serial file.           [2]

|  |
| --- |
| * faster access because applying hashing function to record key (card number) generates physical record address, a mathematical operation which can be computed quickly * there may be many customers and storing the records in a serial file will lead to slower performance and long wait times as the record has to be searched for in a serial file sequentially |

**(ii)** Using the information above, devise a hashing algorithm for accessing   
 information from the file, assuming that there are never more than 10000 cards in   
 use.                             [2]

|  |
| --- |
| * using modulo arithmetic for the hashing algorithm eg mod n where n is a prime number larger than total input size to maximize uniform spread and minimize collision eg 10001 |

**(iii)** State two card numbers which would cause a collision using your algorithm.   [1]

|  |
| --- |
| * 120001 and 130002 will give a collision as both will generate the same remainder of 9990 |

**(iv)** Describe two methods for dealing with collisions.                                           [4]

|  |
| --- |
| * linear probing - store the collided record in the next available position in the prime area (quadratic probing) * store the collided record in the next available position in a separate overflow area * each prime area location will have its own overflow list * devise a new hashing algorithm taking into account the input characteristics to minimize collisions |

**(b)** With reference to the data that can be collected by this system, discuss the effects   
 that holding such data can have for the

**(i)** customers                                                                                                         [3]

|  |
| --- |
| * easy form of identification of member * able to enjoy personalized member benefits/privileges * some users may not be comfortable with disclosing restricted information with some of these stored publicly in the card number eg age |

**(ii)** management                                                                                                     [3]

|  |
| --- |
| * gather accurate picture of profile (gender/age group) of customers attending the type of shows * better target marketing/publicity efforts towards a particular gender/age group * cater to interests/demands/needs of dominant user group * need to put in place measures to protect the personal data collected for privacy protection * adhere to legislation for data protection act |

of the complex.

**2007P2**

**5 (a)** With reference to the details about the customer loyalty card given in question 4a,   
 state two ways in which the rules for determining the card number may need to be   
 changed, justifying your answers.   [4]

|  |
| --- |
| * age is a field value that will only be valid for the current year, a better alternative will be the use either the year of birth eg 1995 or more specifically the date of birth eg 19951225 * 3 digits will be too limited/narrow a range as it can only accommodate a maximum of 1000 male + 1000 female customers, a better alternative will be to use more digits eg 5 to 6 to allow for growth expansion * may need to change age component as info will only be valid for the current year. solution - store year of birth instead of age * membership growth may exceed 3 digits. solution - expand to more digits eg 5 or 6 based on historical statistics or projection of future trends/demand |

**(b)** State what is meant by backing-up a file and describe a suitable back-up procedure   
 for the customer details file.                                                             [6]

|  |
| --- |
| * Backing up a file involves copying the contents of a given file to another file on disk where it will remain untouched until the main file is changed beyond recognition or destroyed. * Computer should perform back-up automatically every five minutes. * Function to back up: * Open main file * Open backup file for writing * for each line in file * read line from main file * write line to backup file * close both files * Main program: * run backup every minute * Using bash: cp ./FILE.DAT ./FILE.DAT.BAK |

**2007P2**

A piece of software is to be written to simulate the evacuation of customers if there is a fire.

**6 (a) (i)** State two items of data which would need to be input to the simulation about each   
 of the theatres.                                                                                               [2]

|  |
| --- |
| * number of exits * seating plan (coordinates of seats, safety/emergency exits, top-down view, side-view) * location of anti-fire equipment/facilities (anti-smoke/fire sprinkles, fire extinguishers) * time of performances (amount of human traffic) |

**(ii)** State two other items of data which would need to be input to the simulation which   
 are not directly concerned with the theatres.                                                       [2]

|  |
| --- |
| * duration/period of simulation * possible location of fire * intensity/coverage of fire * nearest fire station |

**(iii)** State two items of data which will be output from the simulation.            [2]

|  |
| --- |
| * optimal number of fire exits * escape route (based on location/intensity of fire) * number/types/strategic locations of fire extinguishers |

The software is written as a set of modules.

**(b)** Explain why modular programming is sensible in this application.                           [4]

|  |
| --- |
| * development and testing can be isolated and facilitated * break down big complex problem into smaller manageable parts * reuse code from standard/custom modules to shorten development cycle * allow different developers to work on different modules * allow prototyping when core modules are ready and others can be stubs * easy to update eg new types of fire extinguishers, alternative escape routes |

**(c)** The modules have all been fully tested and work perfectly. However, the program   
 fails to run.

Explain why this might happen.                                                                               [3]

|  |
| --- |
| * program may have passed unit testing but not integration and system testing * interfaces/dependencies between modules are not tested properly/fully eg output format from one module is incompatible with input to receiving module * algorithm may not be efficient enough to cater to volume data, hence slow processing leading to illusion that program fails to run |

**(d)** An alternative solution is to write the software using an object-oriented language.

Explain why object-oriented programming would be suitable type of programming. [4]

|  |
| --- |
| * allows concurrent and distributed control/processing, different simulation models can be run * more natural/direct mapping of classes/objects to entities in problem domain eg Theatre is a class with data NumberOfSeats and method Evacuate() * sandbox security protection (encapsulation) so that private data are protected from unauthorized access/modification and only accessible via public methods * inheritance allows code reuse eg common data and methods in superclass FireExtinguisher and subclasses different types of fire extinguishers eg CO2Extinguisher * polymorphism allows concise and powerful mechanism to invoke different classes’ methods of the same name e.g. different Evacuate() methods for different theatre types * event-driven model by message passing |

**SP9754P1**

**2.** Different computer applications may require different file organizations.

**(a)** State how records are organized on:

**(i)** serial files

|  |
| --- |
| * records are not stored in key order but arranged/stored in chronological order as and when they are entered |

**(ii)** sequential files

|  |
| --- |
| * records are arranged/stored in record key order eg by employee id/no |

**(iii)** random files.

[3]

|  |
| --- |
| * records are arranged/stored according to hashing function applied to record key which then gives the physical record address |

**(b)** A large bank uses a computer system for managing customer accounts. For each of   
 the file organisations in (a) give an appropriate use in this application and justify your   
 choice.   [6]

|  |
| --- |
| * serial - bank transaction file contains raw transactions records done by customers during the day to serve as an audit/log file to verify that such transactions have taken place * sequential - sorted transaction file by customer id/no so that batch processing of master file can take place at the end of a month to generate monthly statements * random - customer file containing customer bank details can be queried at any time to look up a particular customer’s balance or transaction information |

**(c)** Explain how a record stored in a random file can be retrieved.                     [3]

|  |
| --- |
| * apply hashing function to record key to generate physical record address in prime area * if record is not found in prime area, then error - no such customer record * if prime area is occupied and key matches, then retrieve customer information * if prime area is occupied and key does not match, access overflow area to search for record sequentially |

**(d)** Explain why the archiving of data is necessary.                                            [3]

|  |
| --- |
| * archival is to store data that is seldom accessed to save storage space * increased file size may lead to slower access speed and performance * there may be a need to reference the data eg for audit or legal purposes * archived data is usually stored offline using cheaper storage media eg tape for cost or offsite for safety reasons |

**SP9754P1**

**5** An Abstract Data Type (ADT) consists of both data type and associated operations.

A linked list ADT has the following operations defined:

(i) Create(x) - creates an empty linked list x;

(ii) Insert(x, item, l) - insert new value, item, into linked list x so that it is at position l in   
 the linked list;

(iii) Delete(x, l) - delete the item at position l in the linked list x;

(iv) Read(x, l) - returns the item at position l in the linked list x;

(v) Length(x) - returns the number of items in the linked list;

(vi) isEmptyList(x) - returns true if linked list is empty.

The linked list is implemented by the use of a collection of nodes that have two parts:   
 the item data and a pointer to the next item in the list. In addition there is a Start pointer   
 which points to the first item in the list.

**(a) (i)** Draw diagrams to show the two different situations that can arise when the 'Insert'   
 operation, specified above, is implemented.                                                 [4]

|  |
| --- |
| * insert to front * insert to middle/end |

**(ii)** Write an algorithm that could be used to implement the 'Insert' operation.   [4]

|  |
| --- |
| **Insert(x, item, i)**  new node n // create new node  n.data = item  n.link = NULL  if i = 1 // insert to front  n.link = Start  Start = n  else // insert to middle/end  curr = Start  loop for (i-1) times // traverse to (i-1)th node  curr = curr.link  n.link = curr.link // new node points to ith node  curr.link = n // (i-1)th node points to new node |

**(b)** Show how to implement the following operations for a stack ADT using the list ADT   
 operations:

**(i)** create new stack;

|  |
| --- |
| CreateStack(s)  Create(s) |

**(ii)** add item to top of stack;

|  |
| --- |
| Push(s, item)  Insert(x, item, 1) |

**(iii)** delete item from top of stack.

[5]

|  |
| --- |
| Pop(s)  if isEmptyList(s) // Length(s) = 0  exit // cannot remove from empty stack  else  Read(s, 1)  Delete(s, 1) |

**(c)** A dictionary ADT is used to store a key value and a definition of that key value.   
 Specify three operations for a dictionary ADT.                                                       [6]

|  |
| --- |
| data = key-value pair  Insert(d, k, v) // insert into dictionary d the entry key k and value v  Delete(d, k) // delete from dictionary d the value corresponding to key k  Search(d, k) // search dictionary d for the value corresponding to key k |

**(d)** State two advantages of using ADTs in program development.                 [2]

|  |
| --- |
| * data/information encapsulation/hiding - interface specifies what operations can be performed on the ADT * code reuse - eg Stack ADT inherits/reuses List ADT operations * implementation independence - need not worry target programming language to implement ADT, focus on high-level logic * changes in implementation will not affect interface and thus client code using the ADT eg search/sort implementation can be changed from linear/sequential/bubble/insertion to binary/quicksort and external programs need not have to be changed i.e. still call the same Search()or Sort() |

**SP9754P1**

**7** A programmer is writing an interactive program to grade exam marks from an   
 examination paper where the maximum mark is 60. The program is composed of a   
 number of modules. One module in the program accepts a mark and determines a   
 grade.

**(a)** Give two other modules that this program might contain. Justify your choice.         [4]

|  |
| --- |
| * input module - get marks from user via keyboard/console * validation module - validate if input is a number/integer in the range 0 - 60, presence check to prevent empty input * display/output module - display the marks with corresponding grade to screen/report |

**(b)** Explain what is meant by

**(i)** white box testing;

|  |
| --- |
| * ensure all possible control paths are exercised/tested and produce expected results (how) * requires explicit knowledge of code within module * usually performed by programmer/developer |

**(ii)** black box testing.

[4]

|  |
| --- |
| * test functionality by providing input-output specification (what) * no explicit knowledge of implementation details, only interface (input and output) * can be performed by system testers (non-technical) |

**(c)** Explain, with examples, the different types of test data that could be used in a full test   
 programme. [6]

|  |
| --- |
| * normal data - data that should be accepted and normal processing performed eg 50 * extreme/boundary data - data that tests against boundary/loop conditions eg -1, 0, 60, 61 * erroneous data - different data type eg abc, out of range eg -30, 100 * volume data - efficiency test to ensure reasonable response time given large input size eg marks of entire class/school provided in text file |

**(d)** When all the modules have been successfully tested name two other tests that the   
 programmer could use.                                                                       [2]

|  |
| --- |
| * beta - performed by a select group of external users, code may contain non-critical bugs, may test for other criteria eg usability/user experience, performance, etc. * integration - test for interface/dependencies between two or more modules (not necessarily entire system) * system - test for all possible interaction/dependencies across all modules in system * user acceptance - guide end-users through typical task scenarios to endorse effectiveness (functional requirements), usability (user-friendliness), performance (efficiency) |

**SP9754P2**

**3** The artefacts are moved regularly. For instance, the sackbut is sometimes displayed as   
 part of the exhibition of wind instruments and sometimes as part of the exhibition of   
 seventeenth century.

Explain how the different types of data can be stored. Your answer should include a   
 reasoned description of suitable storage devices, suitable data structures, and details of   
 how the data is linked.                  [8]

|  |
| --- |
| * artefact information may be artefact id, name, description, year, location * use random file organization to store artefact information to facilitate lookup query and efficient retrieval * use sequential file organization to arrange artefact information by artefact id to facilitate inventory stock take monthly/quarterly/yearly * store artefact information in a relational database for data consistency, minimize data redundancy and data independence artefact info/details can be stored on read-only media that need not be changed eg optical media such as CD-ROM or DVD-ROM * artefacts information can also be stored on solid state device eg flash drive - stable and portable to facilitate movements across departments and outside capital * position/location information can be stored in networked databases or files - need to be updated constantly as artefacts are moved * random file organization for one-time lookup of artefact information - requires direct/random access independent of the collection size * sequential file organization for artefact inventory stock take at the end of each month/quarter/year |

**SP9754P2**

**4** Entry to the museum is through a set of computer controlled doors. The environment   
 inside the museum is air conditioned and the humidity is controlled, making it necessary   
 to have the doors shut as much as possible.

**(a)** Describe the hardware necessary to allow the computers to collect information to   
 allow the safe operation of the doors. Your answer should refer both to times when   
 the museum is open and closed to the public.                                                         [4]

|  |
| --- |
| * proximity sensors to detect movement/presence of people/objects on both sides of door * safety pressure sensors to stop doors on closing on people/objects * electronic locking system when museum is closed * keypad/card swipe/biometric authentication system to manually override if other computerized aspects malfunction |

**(b)** Using pseudocode, or otherwise, devise an algorithm to control the automatic doors.

  [10]

|  |
| --- |
| Algorithm to control automatic doors  Assumption: museum is open 7 days a week during extended hours eg 9 am to 9 pm  ControlDoor  probe time of day  if opening hours  loop keep doors closed  until proximity sensor detects presence of object  open door  loop keep doors open  until proximity sensor detects absence of object  close door  alarm if door cannot be closed  else // museum is closed  loop keep doors closed  until manual/pin/biometric mechanism activated  open door |

**5** The museum sells a full range of goods through its shop. The shop also runs a site on   
 the Internet allowing customers to purchase items remotely. Customers can see a   
 catalogue of goods for sale and order items from it from their remote locations. Payment   
 is normally made by the customer forwarding their credit card details.

**(a)** Describe problems that could arise for both the customer and the museum using this   
 method of payment.                                                                                         [2]

|  |
| --- |
| * customer credit card details could be intercepted by hackers * museum is unable to verify true identity of credit card user for online remote transactions * credit card is charged but goods not delivered |

**(b)** Describe methods that could be used to overcome the problems.           [6]

|  |
| --- |
| * two-factor authentication eg send sms to user to enter auxiliary pin number to confirm/complete transaction * ensure encryption of credit card information and that transmission is over secure channel eg https * do not dispatch goods immediately but await clearance from card company |

Details of the orders are matched against the stock file to check for availability of   
 items before being stored in a transaction file for future processing. Credit card details   
 are processed immediately.

**(c)** Draw a system flow diagram to explain the flow of data through this system.         [6]

|  |
| --- |
| * at least 3 correctly labelled processes * at least 2 correctly labelled input / manual operations * at least 2 correctly labelled output * at least 2 correctly labelled storage * completeness |

**(d)** Explain how the stock file is updated from the transaction file if the stock file is held   
 as a:

**(i)** random file                                                                                                 [2]

|  |
| --- |
| * apply hashing function to record key to obtain physical record address * search for record in prime or overflow area * reduce quantity on stock file from transaction file * if not found, flag error |

**(ii)** sequential file                                                                                             [6]

|  |
| --- |
| * sort transaction according to stock id/no * so that the transaction file record order matches the stock file record order * read and compare first records from transaction file and stock file * if same, update stock record with transaction file details and write record to new stock file * if different, write current stock file record to new store file, read next record from stock file until stock id/no matches * read next record from transaction file and stock file * repeat until end of transaction file * write remaining stock file records to new stock file |

**SP9754P2**

**6** The museum collects funds from many sources including entrance fees, purchases from   
 the shop and restaurants, and donations. It also has to produce the payroll for staff   
 salaries and the payments to suppliers.

Each of these financial applications is currently controlled by a separate system.   
 The museum's board of governors has decided to commission a software house to   
 produce an integrated package to handle all financial transactions.

**(a)** Explain why local and global variables will be important in the production of the   
 software.  [4]

|  |
| --- |
| * software will be written in modules * some common variables may be required in many modules and need to exist throughout lifetime of program * thus using global variables will be appropriate without the need to pass these everytime as parameters to the modules * also large data structures eg array of artefacts info which will be commonly accessed can be global variables to save storage space without it being duplicated unnecessarily using pass by value * within modules, some variables may be needed for intermediate processing * these need not be accessed by other modules, so using these as local variables will help prevent accidental modification * because local variables only exist within the program block in which they are declared, memory may be released when these are no longer needed hence reducing the overall storage footprint of the program |

**(b)** Describe how the software house will test the package.                             [6]

|  |
| --- |
| * white box - test to exhaust all control paths within a module, knowledge of internal logic of code (how) * unit / black box - test for expected output given well-defined input, no knowledge of internal code (what) * alpha - within developer team for critical bugs * beta - for selected group of end users for usability/stability * integration - modules are combined and tested as a group (not necessarily entire system), test for interface/dependencies between modules * system - entire system with all its interfaces and dependencies is tested * regression - when new tests are introduced, old functionality should not be disturbed, ideally automated * volume/performance - ensure algorithms are efficient and return within reasonable response time for large input data sets * user acceptance - end user is guided through a typical process cycle with representative samples to verify that system adheres to functional requirements and meet other expectations of project |

**(c)** The solution contains a number of modules. Three of the modules are:

* + A module to input relevant data to the system
  + A module to calculate the pay for individual workers
  + A module to output payslips and electronic transfer of pay to workers' bank accounts

State the parameters that would be passed between

**(i)** the input module and the pay calculation module giving a reason why each   
 parameter is necessary,

|  |
| --- |
| * worker identification number - to ensure correct worker record is accessed * hours worked - to compute pay |

**(ii)** the calculation module and the output module giving a reason why each parameter   
 is necessary. [6]

|  |
| --- |
| * worker identification number - to identify worker details (eg name) for payslip * pay amount - to output correct payment * bank account details - to ensure correct bank account is credited |

**SP9754P2**

**7** Each item offered for sale in the museum shop has information stored about it in a stock   
 file. Each record in the file contains a number of fields.

**(a)** The description of the item is stored in a fixed length field using ASCII characters.

**(i)** Explain what is meant by the term fixed length field.

|  |
| --- |
|  |

**(ii)** Describe how characters are stored in ASCII format.

[4]

|  |
| --- |
|  |

**(b)** The field which stores the number of items in stock is stored as one byte binary   
 integer.

**(i)** Explain why character and floating point representations would not be appropriate   
 for this field.    [2]

|  |
| --- |
|  |

**(ii)** Describe a situation which would cause the suggested representation to fail and   
 state how the problem could be overcome.                                                       [2]

|  |
| --- |
|  |

**(c)** The personnel records for those employees who work in the shop are stored in a   
 linked list in alphabetic order of surname.

**(i)** Using a diagram, show the list for the employees

Smith, Wu, Arma, Hon

(You should include reference to free space in your diagram)

|  |
| --- |
|  |

**(ii)** Using a second diagram, show how Thor would be added to the list.

[6]

|  |
| --- |
|  |

**8** Items sold in the museum shop are arranged in groups. For example, all items of   
 confectionery are coded 01, while souvenir stationery are coded 02. all books are coded   
 03, and so on. There are a total of 25 such categories numbered 01 to 25.

The computer system is designed in such a way that the total value of each type of item   
 sold during the day is stored in a two-dimensional array containing 25 records, each   
 having two fields, namely

* The name of the category e.g. confectionery, stationery...
* The total value of those items sold that day.

When an item is sold, the assistant at the checkout inputs the two digit   
 number corresponding to the category of item, followed by the cost of the item.

Write an algorithm which is to be used at the end of each day's trading to print out   
 the category codes, in order, according to the value sold, starting with the category that   
 sold most that day.     [6]

|  |
| --- |
|  |

# Part II: Interfaces and Interactions

## Chapter 5: Interacting with Computers

## Chapter 6: Interfacing Computers

## Chapter 7: Interacting with Data

**2014**

**3.** A network manager for a sales company types the following into his computer:

copy C:\monthlysales\\*.dat E\:\backup\junesales /V

**(a)** State the type of user interface being used. [1]

|  |
| --- |
|  |

**(b)** Describe, using the example, two benefits of the user interface named in **(a)**. [4]

The network manager has a disabled user who cannot use a keyboard but can control   
 a point-and-click device that moves a pointer on the screen.

|  |
| --- |
|  |

**(c)** Describe a user interface that would allow this user to enter text into a word processor.   
 [3]

|  |
| --- |
|  |

**(d)** The sales company provides a special user interface for this user. State **two** benefits to   
 the company. [2]

|  |
| --- |
|  |

**2014**

**4.** A small local area network (LAN) in a school consists of one switch, one file server and   
 ten computers.

**(a)** Explain **why** circuit switching could be used in this LAN. [2]

|  |
| --- |
| Circuit switching: Means one to one connection, where the communication line is dedicated and cannot be interrupted.  In this context: Administrator could be using circuit switching to do important server maintenance, so other people cannot use the server at the same time. |

The network has a connection to the lnternet added.

**(b)** Explain **how** packet switching is used when a web page is downloaded from the   
 Internet. [3]

|  |
| --- |
| A user request a page, which includes test, photos and other assets from the server.  The server **breaks up** the page to multiple packets with each packet containing **‘control data’** : which include destination ip address, source ip address(for security and multiple sources/ tabs open), ordering/sequence number of packets (to reconstruct the document) and lastly parity bits and error correction.  These packets then travel in multiple routes to the target location. The user computer reconstructs the packets into a document, and if there is a error, request for the page again or correct the error, if not acknowledge to the server that the transfer is successful. |

A packet from the web server consists of 256 bytes. One of the bytes is the checksum   
 byte. In each byte one bit is the parity bit.

**(c)** If the byte 0 0 1 1 0 1 0 1 results in a parity error, state the type of parity being used.[1]

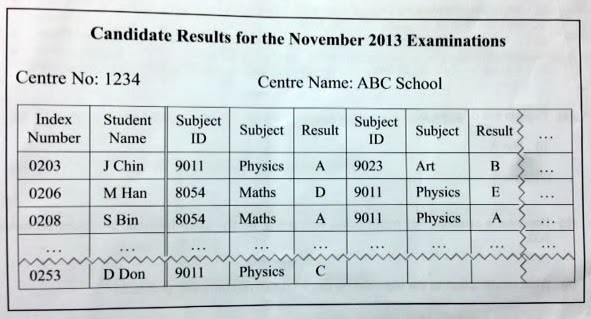
|  |
| --- |
| Odd parity. |

**(d)** The receiving computer uses the checksum byte to check whether the packet   
 contains an error. Explain how it does this. [4]

|  |
| --- |
| The checksum is the ‘**control total’** of the sequence of bytes for the purpose of detecting errors which may have arose from transfer or storage.  Apply the checksum algorithm onto the data received, check if the **generated checksum** is equal to the checksum received. **If it is equal, there is likely no error, else an error had occurred in transmission and a retransmission is needed** (because checksum is not enough to recover the data). |

**2013**

**2** Examination centres receive examination results for their candidates as a printed report.   
 The report lists the candidates in order based on their Index Number. For each   
 candidate their results occupy one row of the report. Each row displays the results for all   
 subjects that the candidate sat in the examination.



Candidates can only take examinations at one centre in a particular session.

Currently the candidate results for each centre are stored in a separate file. The   
 software that produces the above report is written in a programming language.

**(a)** Describe, using an example, why this file has data redundancy. [2]

|  |
| --- |
| Each subject ID has a corresponding subject name.  In each row, both subject ID and subject (name) are provided, and this is repeated across the rows, for the same subject IDs.  Index 0206 and 0208 both have subject ID 8054, and Subject Math.  Data is repeated, hence data redundancy arises. |

**(b)** An extra field is added to the file, but the report will not include this new field.

Describe the problem that will arise. [3]

|  |
| --- |
| The data in the field will be read into the program, which may result in it populating existing fields.  This could lead to wrong processing and/or wrong output (incorrect alignments).  This can also lead to a parse error in the program. |

A normalised database solution to this problem is designed, which has a number of   
 tables.

**(c)** Draw an E-R diagram that shows these tables and the relationships between them.

[5]

|  |
| --- |
|  |

**(d)** When the data are stored in a database, privacy is of great concern.

Explain why. [2]

|  |
| --- |
| Data stored in a database are generally sensitive personal information, such as password, date of birth and contact details.mm  Data privacy is the security measures taken to prevent and defend against database attacks.  If data privacy is compromised, non-authorised parties will be able to gain access and retrieve such sensitive data, potentially misuse/abuse them. |

**2013**

**4** A software development company currently hosts its own email server. The company is   
 considering a replacement webmail service, using cloud computing.

**(a)** State two advantages of this change. [2]

|  |
| --- |
| * easily accessible from multiple computing devices (desktops, laptops, tablets, smartphones) as webmail is a SaaS (Software as a Service) that only requires users to have Internet access to a cloud service. * cost efficient in terms of acquisition and maintenance of infrastructural (hardware servers, software, manpower eg system administrator) savings and allows service to scale when company expands * efficiency of system (hardware and software) upgrades and security patches, ease of setup, no/minimal downtime in maintenance |

**(b)** State one disadvantage of this change. [1]

|  |
| --- |
| * security and privacy * less customizability |

The company is also considering other uses of the cloud. These include collaborative   
 activities between employees of the company and also assistance in developing new   
 software.

**(c)** Describe an example of how employees of the company may use the cloud to work   
 collaboratively. [3]

|  |
| --- |
| * type of service (document editing or spreadsheet, presentation) and purpose * multiple location, different devices, timezone, centralized document * automatic notification, revision history, ease of sharing with collaborators, different access rights (edit, comment, view), online chat or discussion |

**(d)** Describe how the cloud can be beneficial to the company when developing new   
 software for a client. [4]

|  |
| --- |
| * PaaS (Platform as a Service) - define * enable different developers to collaborate on same code base, efficient version control (commits, pull requests) * allows for scalability of application due to cloud usage patterns eg on and off, growing fast, unpredictable bursting, predictable bursting * developers can focus on application and data |

**SP9597**

**4** An email packet consists of 128 bytes. The first 126 bytes contain both control data and   
 also a part of the email message. Byte 127 in the packet is a checksum. Byte 128 is   
 currently not used. Even parity is used in each byte.

**(a)** If the first seven bits of a byte are 0 0 1 1 0 1 1 state the value of the parity bit. [1]

|  |
| --- |
| * 0 |

**(b)** A computer receives a packet which contains the following byte: 0 1 1 0 1 1 1 0.   
 Describe the actions of the computer upon receipt of this byte. [3]

|  |
| --- |
| * check that parity of byte is OK * odd number of 1s - problem with parity * issue request to server to retransmit byte / report error |

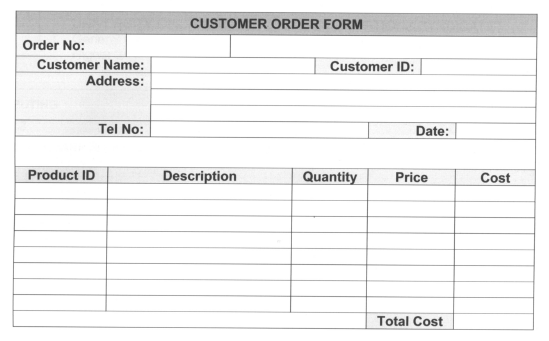
**(c)** Describe how the checksum byte is produced by the sender computer. [3]

|  |
| --- |
| * the first 126 bytes are * put through an algorithm that * adds the contents together (possibly with positional weightings) * any overflow is discarded so that result is one bye value   1 mark per point to a max of 3 |

**(d)** It is decided to use the unused byte as a parity byte. Explain, using an example, how   
 this byte could be used to detect and correct a single bit error in the packet. [3]

|  |
| --- |
| * each byte already has its own parity bit (vertical parity) * unused byte can be a collection of parity bits for longitudinal parity * so first bit is parity bit for all the first bits of the 127 bytes * if single bit error then the two incorrect parity bits will pinpoint error * bit can be automatically corrected (0 becomes 1 or 1 becomes 0)   1 mark per point to a max of 3 |

**SP9597**

**6** An order form used by the customer of a supplier of goods has the following form:

* Each customer may send many orders.
* Each order may have multiple lines for products.
* One line contains the details for the order of an individual product.
* A product can appear only on one line on an order.
* A product may occur on many different orders.

**(a)** Identify the tables that will give a normalized solution to this problem. Draw an E-R   
 diagram that shows these tables and the relationships between them. [6]

|  |
| --- |
| Alternative methods of showing the type of relationship (e.g. writing 1:n) are acceptable   * 1 mark for tables Product, Order, Customer * 2 marks for table Orderline * 1 mark for correct Product-Orderline relationship * 1 mark for correct OrderLine-Order relationship * 1 mark for correct Order-Customer relationship |

**(b)** A table description can be expressed as:

TableName(Attribute1, Attribute2, Attribute3, …)

The primary key is indicated by underlining one or more attributes.

Give the table descriptions for the tables. [10]

|  |
| --- |
| * Customer holds the details of customers only - Customer(CustomerID, Name, Address, Telephone, …) * 1 mark for the key and 1 mark for two fields * Product holds details of each product - Product(ProductID, Description, Price, ...) * 1 mark for the key and 1 mark for two fields * Order holds enough information to connect a customer to an order - Order(OrderID, CustomerID, OrderDate) * 1 mark for the primary key, 1 mark for the foreign key and 1 mark for the OrderDate * OrderLine(OrderID, ProductID, Quantity) * 1 mark for each correct part of the compound key and 1 mark for the quantity |

UNF

PatientID, PatientName, Address, HomeTel, Mobile, NOK, DoctorID1, DoctorName1, AppointmentDate1, AppointmentTime1, Description1, DoctorID2, DoctorName2, AppointmentDate2, AppointmentTime2, Description2, ...

1NF (remove repeating field)

PatientID, PatientName, Address, HomeTel, Mobile, NOK,

AppointmentID, Patient ID, AppointmentDate, AppointmentTime

AppointmentID, DoctorID, DoctorName, Description

2NF (remove partial key dependency - doctor name is not dependent on appointment ID)

PatientID, PatientName, Address, HomeTel, Mobile, NOK

AppointmentID, Patient ID, AppointmentDate, AppointmentTime

AppointmentID, DoctorID, Description

DoctorID, DoctorName

3NF (remove non-key dependency)

* Patient(PatientID, PatientName, Address, HomeTel, Mobile, NOK)
* Appointment(AppointmentID, PatientID, AppointmentDate, AppointmentTime)
* AppointmentLine(AppointmentID, DoctorID, Description)
* Doctor(DoctorID, DoctorName)

**2012P2**

**6** The head office of the transport ministry has a number of different departments. Two are the   
 finance department and the administration department

The finance department has a number of computers arranged in a network with a bus   
 topology, whereas the administration department has a computer network arranged in a star   
 topology. The computers in the administration department share resources.

**~~(a)~~** ~~By drawing a diagram or otherwise, describe a star topology. [3]~~

**(b)** Explain the purpose of each of the following in a network and state how it could be used at   
 the transport ministry.

**(i)** Server



|  |
| --- |
| * Host machine which processes requests and transmits data to other computers in the network. Also stores files, applications and services, allowing computers to access them via clients. * Used to store financial records in finance department, vehicle owner data in the administration department. Processes the requests for files and delivers them to the respective users who requested them |

**(ii)** Switch

|  |
| --- |
| * Hardware which **connects multiple computers**, printers and servers within the same Local Area Network * Able to inspect incoming data packets, recognise their origin and destination and process them accordingly, allowing for higher efficiency and lower bandwidth * Enables computers in each department to communicate with each other, share information and allocate resources efficiently |

**(iii)** Bridge

[6]

|  |
| --- |
| * Connect multiple **networks** together * Passes messages between **two segments** of a Local Area Network, in the Transport Ministry, between the finance department and administration department, allowing them to share files with each other if needed. E.g. finance department send financial records to administration department to process and generate payroll * Can be used to segment the bus network in finance network into two segments to improve performance and minimise collision, prevent unauthorised access |

**(c)** Data sent around the network are sent using even parity.

Explain what is meant by even parity and state why it is necessary. [4]

|  |
| --- |
| * In even parity, ~~the left-most~~ one bit is used as a parity bit to ensure that the number of 1s in the data is even as a form of error detection mechanism. The parity bit will be 0 for data which has an even number of 1s, 1 for data which has an odd number of 1s. * Even parity is necessary to detect whether the data sent is the intended, correct data, because there might be transmission errors occurring when sent through the network, or interception by external parties (hackers), causing the number of 1s to become odd if one of the bits is changed. Errors with 2 or more bits (in even numbers) changed cannot be detected |

**2011P1**

**3.** A school currently stores their data in flat files. These files contain information about teachers,   
 students and classes.

**(a)** Using examples from the school’s flat files to illustrate your answers,

**(i)** explain how data inconsistency could arise, [3]

|  |
| --- |
| A student may have a class and a teacher assigned. However, if the student changes class, the assigned teacher may not have changed, leading to an inconsistency of data in the form of an update anomaly. If a student does not have a class yet, perhaps due to classes not assigned during orientation, there may be empty fields in the flat file, leading to inconsistency of data in the form of an insert anomaly. When students graduate from the school, their names may be deleted from the file, but it could be along with other data such as teachers; if the teacher only teaches one class, deletion of the students (assume that the whole class graduates), may lead to deletion of data for the teacher, leading to inconsistency in the form of a deletion anomaly. |

**(ii)** explain why data could be redundant. [3]

|  |
| --- |
| In a flat file, the teachers’ data would be stored together with all the students and classes. However, the teachers’ data is dependent on the teachers’ names. This means that repeated data would be stored when a class is stored multiple times, leading to redundancy of the data. |

The decision has been made to create a database with four tables: student, class, student-class, and   
 teacher.

**(b)** Explain why the use of a database will provide data independence. [3]

|  |
| --- |
| Data independence is when modification of the data’s definition does not affect other schemas. Changing how data is defined in a flat file would require an update to other features such as the code accessing the data. The database management system would interpret how we want the schema to be, whereas the controlling mechanism remains independent, therefore providing data independence. |

Each student is in a number of classes. A teacher can teach a number of classes but a class   
 has only one teacher.

**(c)** Draw an E-R diagram to show the relationships between the four tables. [4]

|  |
| --- |
|  |

**(d)** For each table specify the attributes (fields) required and state suitable primary keys. [10]

|  |
| --- |
| Student(Name,StudentNRIC,DateOfBirth)  Student-Class(StudentNRIC\*,ClassName\*)  Classes(ClassName,TeacherNRIC\*)  Teacher(TeacherNRIC,Name,Salary) |

**2011P2**

**3.** Fish4T use computerised point of sale terminals which are networked within each shop. The office   
 building has two networks, one used by the office staff and the other by the accounts department.   
 There is also a computer system in the warehouse. The individual systems are all connected together   
 to form a network and this network has Internet access.

**(a) (i)** Describe the characteristics of a local area network (LAN). Identify a LAN used by Fish4T. [3]

|  |
| --- |
| network covering small specified geographical region  uses physically wired connection/cabling to ensure high bandwidth transmission  information shared and stored is protected by firewalls set up by network administrators  The network used by the office staff |

**(ii)** Describe the characteristics of a wide area network (WAN). Identify a WAN used by Fish4T. [3]

|  |
| --- |
| covers larger geographical area than Local Area Network  consists of two or more LANs linked together  mixture of wired and wireless communication technology  information stored is generally insecure  The Internet to which the LANs of Fish4T is connected |

**(b)** Explain the purpose of each of the following network components. State where each would be used   
 in the Fish4T systems.

**(i)** Server [2]

**(ii)** Router [2]

**(iii)** Bridge [2]

|  |
| --- |
| Router:  device that forwards data packets along networks based on the IP(Internet Protocol) address of the recipient  used to connect multiple networks  Connect the network used by the office staff and the network used by the accounts department to the internet? |

**(c) (i)** Describe the consequences of the computer systems failing. [2]

|  |
| --- |
| Situations of failure >> consequences  Possible situations of computer systems failing include a faulty wire, internet cable, or simply corruption of the hard drive of the computer.  Customers’ orders would be lost if data is not backed up. Additionally, security systems may be compromised during the failure, causing plausible leakage of private customers’ information. If the sales are recorded through the system, the company will be unable to sell fish from the warehouse, resulting in a loss of revenue and possible customer dissatisfaction.  Office workers will not be able to work.  Accounting worker cannot do accounting. |

**(ii)** Describe the measures that can be taken to limit the effects of failure of the computer systems. [4]

|  |
| --- |
| To prevent human negligence, keep a checklist on the wall for technician staff to checkoff and ensure the system is running smoothly, i.e. any breach of network, power levels, data integrity etc.  To ensure that the data is recorded and kept safe, backup customer orders using a Redundant Array of Inexpensive Disks system / cloud storage services such as Google drive to ensure that any customer data will not be lost or corrupted as verification can be done with the backup data to restore the original version properly.  To maintain customers’ orders even during downtime, implement a contingency plan i.e. a backup manual system to take over in case of failure when some time is needed to fix the problem. For example, open fish orders can be stored inside a physical cabinet to take orders physically on paper from customers to be keyed into the database later. Additionally, customers may be allowed to make phone calls to make orders.  Ensure constant saving of data through automated, consistent background backups. To ensure a backup power supply, it is possible to include a backup generator or a Tesla powerwall. Solar panels can also be installed to ensure constant power supply off the power grid. If all else fails, employees can run on a treadmill to provide minimal power to run key aspects of the system.  If there is hard drive corruption, use software to reconstruct the data in the hard drive. |

**2010P1**

**5** The present record keeping system is stored in flat files on each of the three computers used by the

employees in the office.

**(a)** It has been suggested that the flat files be used to create a relational database containing a   
 number of tables.

Two of the tables in the database will be the APARTMENT table and the MAINTENANCE table.

**(i)** Explain the advantages of using a database rather than flat files. [4]

|  |
| --- |
|  |

**(ii)** Draw the relationship between APARTMENT and MAINTENANCE in the form of an ER   
 diagram. [1]

|  |
| --- |
|  |

**(b)** State the meaning of each of the following, giving an example of it from the MAINTENANCE table:

**(i)** primary key, [2]

|  |
| --- |
|  |

**(ii)** secondary key, [2]

|  |
| --- |
|  |

**(iii)** foreign key. [2]

|  |
| --- |
|  |

**(c)** The three computers in the office will be networked. It is decided to put an additional computer in   
 the maintenance workshop. This computer will be connected to the network.

**~~(i)~~** ~~The network will be based on a bus topology.~~

~~Draw a diagram to show the bus topology. [2]~~

**(ii)** Explain why a protocol is needed. [2]

|  |
| --- |
| * a protocol is a formal set of rules and conventions that governs how computers exchange information over a network medium * protocols are necessary as there are different types of computers, hardware and networks * therefore there need to be rules that specify how these differences are to be resolved if communication is to be possible |

**(d)** When data is transmitted across the network it is sent in bytes. The following bytes of data have   
 been received by a device on the network.

01101101 10110100 01101000 10100001

One of the bytes has been corrupted.

**(i)** State which is the corrupted byte, justifying your choice. [3]

|  |
| --- |
|  |

**(ii)** Explain how transmitting bytes in blocks can allow the receiving device to self-correct errors. [2]

|  |
| --- |
|  |

**2009P1**

**4 (a)** Explain why using a database, rather than flat files, results in:

**(i)** improved data consistency, [3]

|  |
| --- |
| * update or delete operations affect all instances of the record in a database as data is linked to each other through different tables. * eg deleting a customer name should delete his contact or records as well(removing a customer from all records) * Same as 2007P1 |

**(ii)** prevention of data redundancy, [3]

|  |
| --- |
| * Databases help to reduce duplication of data as it stores data within a relational database structure, where data is linked to each other through different tables * Eg: customer information need not be unnecessarily duplicated in the orders table but can be referenced in the customer table via the customer id which serves as the foreign key. This stores the minimum amount of information that needs to be stored. |

**(iii)** data independence. [3]

|  |
| --- |
| * each database table should contain information about one entity * eg customer table contains only customer id, stock name, contact no; stock table contains only stock id, stock name, description, price; order table contains order id, customer id, stock id, quantity * allow reuse of database table eg inventory application can make use of stock table * transaction only needs to update one relevant table and not affect other tables |

**(b)** What is a database management system (DBMS)? [2]

|  |
| --- |
| * software to manage a database. A database is a organised collection of data, organised in tables. Row is a record, column is a field. * Software to allow creation, definition and manipulation of database * includes facilities such as   + table structure specification (field name, field type, field length, primary key, secondary index, etc.)   + insert, update, delete and search for records (CRUD)   + authentication mechanism eg username and password to authorize access   + easy to use query facility eg search for subset of records fulfilling a given set of criteria (eg top results in descending sort order)   + report generator to produce output organized by groups (eg sections, subtotals and summary) |

**(c)** By using an example, explain how a query language can aid users to extract data from a    
 database. [3]

|  |
| --- |
| * user can use SQL (Structured Query Language) to query a database given a set of criteria * eg to search for results of students who failed * general format of a query:   SELECT <fieldname(s)> FROM <tablename(s)>  WHERE <criteria/condition>   * SELECT student\_name, marks FROM students, results   WHERE students.student\_id = results.student\_id AND results.marks < 50 |

**(d)** Describe another tool that users can employ to make use of data in a database. [3]

|  |
| --- |
| * report generator * organize query results and generate output in a form that is presentable for user viewing and decision making * eg report layout contains header, section, grouping by subtotals, overall total, summary, footer |

**2009P2**

**5**  The company has a warehouse, two shops and an office building.

Each of the four buildings has a local area network (LAN). The office building has two separate LANs   
 for the accounts office and the administration office. The company has decided to connect the four   
 buildings into a wide area network (WAN).

**(a)** State **two** differences between a LAN and a WAN. [2]

|  |
| --- |
| * LAN is within a small geographical location; generally all physically wired connections. * WANs are generally over large distances; uses large variety of connections such as analogue, wireless and wired requiring signal converters. |

**(b)** State how each of the following is used in a network. Give an example of the use of each in this

application:

**(i)** file server [2]

|  |
| --- |
| * file server is a central repository that holds data/files/programs that the network needs to access regularly and remotely. Large files or commonly shared data/files/programs are best centralised in the server for convenience and efficiency. |

**(ii)** router [2]

|  |
| --- |
| * router connects local area network (LAN) to wide area network (WAN) to internet |

**(iii)** bridge      [2]

|  |
| --- |
| * bridge connects/segments two parts of a LAN together or bridge LAN and WAN |

**(c)** The data is subject to corruption when it passes from one device to another.

Describe how a checksum can be used to detect such errors. Give an example of when it might   
 be used in this application. [3]

|  |
| --- |
| * sender compute check sum of data using some encrypted aggregate function on data eg MD5 or SHA1 * sender transmit data and check sum separately to receiver * receiver run same check sum generator on data * and verify that both transmitted and generated check sums matches * if a data file is corrupted, small variances produces large differences in check sum * eg software programs distributed via the Internet often comes with a check sum for the receiver to verify the file’s integrity   note: check sum != check digit (usually computed and sent as part of data, not separately) |

**(d)** The company uses its own intranet for communication between the buildings.

Explain why this is preferable to using the internet. [4]

|  |
| --- |
| * more secure transmissions since intranet is usually protected by firewall * easier dissemination of stock data * intranet are usually LANs thus they are physically hardwired allowing faster transmission time and less transmission errors * easier to control access to the intranet since intranet is protected by firewall and will require controlled login |

**2008P1**

**~~1. (a)~~** ~~Draw diagrams, including servers and clients, to show the three basic network topologies:~~

**~~(i)~~** ~~bus;~~

**~~(ii)~~** ~~star;~~

**~~(b)~~** ~~Give one advantage and one disadvantage of each topology.~~

**(c)** Explain the purpose of the following items of network hardware. For each item give an example of   
 how it could be used in a network.

**(i)** switch

|  |
| --- |
| * Switch connects two or more devices together and forwards transmissions between devices at full bandwidth by switching between devices. Generally it is connected between a client and a server on a bus network to make a more robust bus network. |

**(ii)** router

[4]

|  |
| --- |
| * A router can be used to connect one or more devices and bridge multiple LAN or WANs while enabling connection to the internet. This can be used to hook up clients at home to the internet through a server on the Internet Service Provider (ISP) end. |

**(d)** Explain why standard protocols are needed in wide area networks.                             [3]

|  |
| --- |
| * a protocol is a formal set of rules and conventions that governs how computers exchange information over a network medium * protocols are necessary as there are different types of computers, hardware and networks * therefore there need to be rules that specify how these differences are to be resolved if communication is to be possible |

**2008P2**

A leisure company owns and operates a bowling alley complex.

**1.** Customers can book a lane by accessing the customer area of the company website.

**(a)** Describe how a new customer could use the Internet to find the company website.     [3]

|  |
| --- |
| * enter company website address at location bar of browser * navigate to the customer area using site's navigation menu or site map, or * use a web browser to launch search engine * enter relevant search/query terms / keywords eg book bowling <leisure company name> * click on retrieved search results showing company link (sometimes directly to customer area) |

**(b)** The customer has to give various details in order to book.

**(i)** Explain why the website might use a forms dialogue interface to obtain the details of the   
 booking.                                                                                               [2]

|  |
| --- |
| * guide customer through booking process to enter relevant information * use of drop-down menus and radio buttons to eliminate the need for validation * user interface controls that adapt to user input * some prerequisite booking info may need to be determined before subsequent steps can follow * eg intended date of booking needs to be provided before information on available lanes for that date is displayed to the user |

**(ii)** State two features, on the form, which may be designed to help the user input data correctly   
 and accurately. Give an example of data which would be input using each feature.                [4]

|  |
| --- |
| * date picker control for intended date of booking * allow user to select a date using a calendar-like interface * prevent free text entry in which invalid date may be entered * leading to error in processing or complicated validation checks * another feature? * number of lanes picker control for intended lanes to book * allow user to select the number of lanes to book * prevent free text entry in which invalid number of lanes might be entered * eg 99 when there is only 12 lanes * time slot picker control where the user can selected the intended time of booking. * prevent free text entry which an invalid time may be entered. |

**(iii)** Explain why a graphical user interface would not be suitable for this application.     [2]

|  |
| --- |
| * GUI is generally well suited for one-time booking on one date and limited number of lanes depending on system design (eg may only be allowed to book up to x lanes) * some less common uses may make this workflow less efficient eg booking on multiple dates, recurrent bookings * an alternative interface (eg upload csv file with booking details) facilitating some kind of batch processing and conflict resolution may be more efficient for such uses |

The company runs a club for people who wish to compete against each other.

**(c)** One piece of information which members should input when booking a lane is their membership   
 number.

**(i)** State what is meant by validation of data input.                                                 [2]

|  |
| --- |
| * validation of data input ensures that input is acceptable/reasonable by conforming to certain rules such as presence, existence range, format, length, data type and other constraints (eg to date > from date) so that normal processing can continue * to prevent erroneous outcomes/results by providing specific error messages to help user recover from error * data input validation - process to determine if data input is accurate, up to date (check for membership expiry), in correct format/range/length for further processing |

**(ii)** Explain why an existence check would be a sensible validation check to use for membership   
 number, while a presence check would not.                                     [4]

|  |
| --- |
| * membership number needs to be checked against database of members * to ensure such a member exists before he/she is allowed to make booking * presence check only detects if an input field is non-empty or satisfied input format * but does not ensure that there is such an existing member, which will cause error in using the membership number as the primary key to retrieve member's particulars * presence check determines only that some input is entered, but not if it is a membership number that exists, which is validated by the existence check. * eg 0000 may pass through presence check, but may not be a valid membership number which starts from 0001 |

**2008P2**

**6.** The data about the club is kept in a database.

The club runs a number of TEAMs.

Each TEAM has a number of MEMBERs playing for it and one or more COACHes.

A MEMBER can only play for one TEAM but a COACH may train more than one TEAM.

**(a)** Draw a fully labelled ER diagram to show how the entities TEAM, MEMBER, COACH are related,   
 while keeping data redundancy to a minimum.                                             [6]

|  |
| --- |
|  |

**(b)** Using examples taken from this application explain what is meant by:

**(i)** a primary key or key attribute;

|  |
| --- |
| * primary key is a field that allows one to identify a particular record uniquely * eg member\_id in MEMBER table |

**(ii)** a foreign key;

|  |
| --- |
| * foreign key is a field which is a non-primary key in a table but a primary key in its own reference table * a non-unique link field to relate one table with its reference table * eg member\_id/coach\_id in TEAM table |

**(iii)** a composite key.

|  |
| --- |
| * formed by joining two or more fields to form a primary key to uniquely identify a record * eg coach\_id + team\_id in COACH\_TEAM\_ALLOCATION table |

**2007P1**

**2. (a)** Explain why using a database, rather than flat files, results in:

**(i)** improved data consistency,           [3]

|  |
| --- |
| With flat files, data is often duplicated. When a modification has to be made to the piece of data (eg A customer changes his mobile number and needs to update it), it has to be updated over each instance of the data. This has a high chance of leading to an instance of the data not being updated, leading to inconsistent data.  In a database, data is not duplicated but are linked to each other through tables. When data is modified within this structure, all other instances are updated simultaneously. This ensures data consistency throughout the database. |

**(ii)** prevention of data redundancy,      [3]

|  |
| --- |
| Flat files store data in a number of files, and data is likely to be duplicated as it is referred to multiple times.  Databases store data within a single normalized relational database structure, where data is linked to points of reference to it. Duplication of data is minimised and thus there is no data redundancy.  Normalised relational databases also removes data dependencies between the data. |

**(iii)** data independence.                      [3]

|  |
| --- |
| Data independence in databases refers to the ability to alter the structure or functions of the database without needing to rewriting software for the entire database. This can be achieved through physical and logical data independence, where physical storage and the logical schema of the database can be altered without having to rewrite the program.  -define data independence between data and dbms. framework in place that allows data to be captured such that both are independent from each other  -dbms interprets what we want the schema to be, the controlling mechanism remains independent of the data  - in database data entities are related to each other, improving data redundancy  -eg of flatfile vs database |

A mail order company plans to create a computer system to store data on:

* stock
* customers
* customer orders

Each order can be for more than one stock item. The details about one stock item on the

order form is called an order line.

A solution is to create a database with four tables: stock, order, order-line and customer.

**(b)** For each table specify the attributes (fields) required and state the primary key for each table. [10]

|  |
| --- |
| stock(stock\_id, stock name, description, price)  order(order\_id, customer\_id\*, delivery\_date)  order\_stock(order\_id\*, stock\_id\*, quantity)  customer(customer\_id, customer name, contact no, address)  \* foreign key   * primary key - uniquely identify a record eg stock\_id in stock table * foreign key - a non-primary key field in the current table but a unique identifier in its reference table eg customer\_id in order table which identifies which customer makes the order * composite key - made up of two or more fields to make the record unique eg order\_id + stock\_id in the order\_line table |

**(c)** Draw an E-R diagram to show the relationships between the four tables.                   [4]

|  |
| --- |
|  |

**2007P2**

A cinema complex has a number of rooms with screens (called theatres), each of which can accommodate a different number of customers. There is also a central area where the booking of seats is done and refreshments are sold.

**1.** The staff dealing with bookings have a computer linked to a network so that resources can be shared.

**~~(a)~~** ~~The computers are arranged using a bus network topology.~~

~~Describe a bus topology and the hardware necessary to operate the network.~~

**(b)** Characters transmitted across the network are subject to corruption.e

Parity checks are carried out to identify errors.

The following bytes

     01010101     10001001     10000001     00111100

are received as part of a transmission.

**(i)** State which one of the four bytes has been corrupted, justifying your answer. [3]

|  |
| --- |
| * second byte 10001001 because it is the only byte that contains an odd number of 1s thus violating even parity error detection scheme * even parity means the total number of 1s within a byte must be even * data/network communication generally occurs in spikes, so likely only one portion of data corrupted |

**(ii)** Explain why there is no certainty that the other bytes have not been corrupted.     [2]

|  |
| --- |
| parity as an error control scheme does not cater for cases when an even number of bits are corrupted eg 01010101 may become 01010110 (transposition error) without being detected |

**(iii)** Explain how parity can be used as a self-correcting mechanism for errors in data transmission     [3]

|  |
| --- |
| * Introduce a parity byte block adhering to even parity   01010101  10001001  10000001  00111100  01110001   * intersection of horizontal and vertical party checks will indicate position of error * since data is in binary, correction is easily done by flipping the bit |

**(c)** The cinema complex is part of a national chain of  cinemas. Communication between all the

cinemas and the chain’s head office is via an intranet.

**(i)** Explain the difference between the internet and an intranet.                        [2]

|  |
| --- |
|  |

**(ii)** Explain why the cinema chain uses an intranet for communications rather than the internet. [4]

|  |
| --- |
| * for a single organization * controlled access for selected group of authorized users eg online sales personnel and staff with accounts * prevent unauthorized access or hacking by hardening security of hardware (server/network) and provision of firewall * facilitate communication and dissemination of internal marketing/publicity materials amongst authorized users |

**SP9754P1**

**6.** A network is to be created in a school with a connection to the Internet. ~~Three network topologies are~~ ~~to be considered:~~

~~(i) ring;~~

~~(ii) bus;~~

~~(iii) star.~~

~~(a) For each of these network topologies draw a diagram to show how clients and servers are~~ ~~connected.~~  ~~[3]~~

~~(b) Give one advantage and one disadvantage of each topology.~~

The network designer can possibly make use of the following network hardware:

(i) switch;

|  |
| --- |
|  |

(ii) bridge;

|  |
| --- |
|  |

(iii) router.

[6]

|  |
| --- |
|  |

**(c)** Explain the purpose of each of these items of hardware. Give one situation where each could be   
 used. [6]

|  |
| --- |
| * switch   + forward messages between computers attached to it   + can be used to make bus network more reliable by attaching clients directly to it and attaching switch to main cable * bridge   + pass messages between two segments of a bus local area network (LAN)   + can be used to segment a bus network into two sub segments to improve performance and minimize collisions * router   + forward messages from one network to another   + used to connect LAN (local area network) to WAN (wide area network) i.e. internet |

**(d)** What is an intranet? Give two ways in which the school might use one.                       [3]

|  |
| --- |
| * private local area network within a small geographical location for an organization * clients and servers are cabled to provide fast, reliable and dedicated bandwidth and access * access is only open to authorized users * school users can house and access school documents (file server) * school users can access online services (eg learning management system, email, etc.) |

**(e)** Describe three possible threats to the network as a result of the connection to the Internet. For   
 each threat give a possible solution.                                                   [6]

|  |
| --- |
| Threats   * virus - install anti-virus software and keep virus definition files regularly updated to constantly monitor/scan incoming network traffic * hacking of servers / denial of service attack - install firewall to monitor and screen off unauthorized network traffic, whitelist authorized users and blacklist non-authorized access * spam - install anti-spam/spam-filtering to block/filter spam messages * spyware - have regular strong password reset policy, anti-spyware program to scan and detect keystroke loggers   malware - virus + spyware  Solutions   * Viruses: constantly updated anti-virus softwares can check system for new viruses * Hacking: firewall to prevent unauthorised access, anti-spyware to prevent leakage, strong password with periodic changes. * Denial of Service(DoS) attacks: ban suspiciously high traffic from an originating IP and limit number of concurrent requests by a single IP address. |

**SP9754P2**

A large national museum maintains a collection of artefacts covering all aspects of the national culture. The museum is split into ten departments. It is estimated that the smallest of these departments has over ten thousand artefacts in its collection.

Seven departments are housed in a large building in the capital, but the other three are in different cities in the country.

**1.** The seven departments in the capital are linked using a LAN, while the other three are linked via a   
 WAN.

**(a)** Explain the difference between a LAN and a WAN.                                                 [2]

|  |
| --- |
| LAN   * covers small geographical area within a specified area * generally limited to a single organization * high transfer speed to transmit information * uses physical cabling to ensure high bandwidth transmission * uses digital and quality of service transmission to ensure high accuracy * information secured with firewalls and accessible only to authorized personnel   WAN   * covers a larger geographical area e.g. across cities/countries * may involve several different local area networks (LANs) linked together * uses a mixture of wired and wireless communication technologies * some parts of data transmitted via traditional analogue lines thus modulation demodulation using modem required * information is generally not secured and non-essential drop rates are acceptable |

**~~(b)~~** ~~Describe the difference between using a bus topology and a star topology on the LAN giving an~~

~~advantage and a disadvantage of each.                                                   [6]~~

**2.** A computerised version of the museum catalogue is to be produced. This will be available to   
 subscribers over an intranet.

**(a)** Explain the advantage of making the catalogue available on an intranet rather than the Internet.

|  |
| --- |
| * faster access due to dedicated physical cabling and higher bandwidth allocation * controlled access to ensure only subscribers can access content and * increased security as only small number of users and connections, malicious access blocked off * easier to charge for services as payments/orders can be handled more easily/safely by setting up accounts * easy to perform audit to measure effectiveness/performance * easy to release exclusive content to subscribers |

**(b)** It will be necessary for the catalogue to allow different methods of searching for specific    
 information.

Describe two ways by which users of the system might be able to search the catalogue of   
 information on a specific topic, giving one advantage and one disadvantage of each method.

[6]

|  |
| --- |
| * free text search   + user enter query/search terms in search box, system performs pattern matching in index and actual text content   + able to search across all text in cataogue   + may return many results but rank relevance may be low   + require user to click through to ascertain relevance, hence more time consuming * browse by category or keywords/tags   + user enter keywords/tags or select from predefined list   + reduces/eliminates input errors and leads directly to information   + can attach rank/relevance to keywords/tags   + may have large number of items hence overwhelming list of choices * menu/tree system   + top level choice leading to successively lower levels   + simple to understand / natural to use / easily applied to mouse/touch screen   + need to understand how information is organized/structured as many layers for large collection |

# Part III: Systems Engineering

## Chapter 8: System Development Cycle

## Chapter 9: Project Management

## Chapter 10: Network Applications

**2014**

**1.** A supermarket chain wants to encourage customers to return to its store. They operate   
 a scheme of rewards for customers based on how much they spend over a period of   
 time.

Customers are issued with a card that is readable by a Point of Sale (POS) terminal.   
 When a customer provides their card at the checkout, the system identifies them and   
 stores the products they purchased and how much they spent.

Currently the only use of this data is to issue the customer with vouchers every three   
 months. Vouchers have a value based on the total amount the customer has spent   
 during the previous three months. The vouchers can only be used in part payment for   
 goods bought in the supermarket.

The supermarket managers want to make more use of the customer purchase data.   
 They hire a software development company to produce software that will implement   
 new uses of the data.

Software developers have skills in developing software. The supermarket managers   
 have in depth knowledge of their business. At first, software developers will have little   
 knowledge of the business.

**(a)** Explain how the supermarket managers can communicate to the software   
 developers what they require. [2]

|  |
| --- |
| (software development lifecycle, do not use technical terms)  The supermarket manager can tell the developer   * when the project needs to finish i.e. time (constraints), * what are the deliverables * project objectives * documentation of the current system of receipts / cashier machines i.e. current input form, output receipts * (can show the developer the process in person, such as a customer purchasing an item. Note: This is not under “communication” and cannot be used in this question.) |

**(b)** Before designing the new software, the software developers need to understand the   
 content and structure of the customer purchase data.

Give **two** methods that can be used for this task, justifying the use of each. [4]

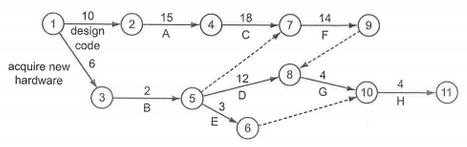
|  |
| --- |
| Content and information   * date and time of the transaction * sales amount * customer name and address (to send vouchers?)   Stored in a relational database / flat file / serial file (but do not write this in the answer)  1) Identify the structure of the data collection by reading existing documentation (how the data is collected and stored) whether it is in a flat file or relational database, so as to determine whether or not the records can simply be called by a key, or require parsing of files to extract data for use.  ~~2) Research possible ways for the completion of the project by conducting interviews with customers to find out how their shopping experience can be improved as to new offers or discounts/sales that would attract them. Additionally, interview the supermarket managers to find out their constraints and possible ideas that they have that they would wish to implement in the supermarket.~~ (WRONG: NOT CONTENT OR STRUCTURE)  3) Look at the receipt to find out which components are most important to customers, such as unit price of the items they bought, the units of the items they bought and the discounts they received in their purchase etc. such as most commonly bought items, price range of items that customers are willing to buy etc. (how can you see this from a receipt?)  4) Attempt to figure out the lack of documentation from the previous developer to figure out the lacking components of the system such as data manipulation etc. so as to implement new/improved ways of system implementation. |

**(c)** Once the analysis phase has been completed, describe what decisions software   
 developers need to make before coding can begin. [6]

|  |
| --- |
| ~~1) Programming style~~ (WRONG: every project should have good programming style; and not specific to this context)  2) Programming stack used, for example programming language, database, operating systems, front end technologies such as javascript.  3) System flow chart i.e. a diagram of the procedures/steps of an algorithm/program, a visualization of what is going on in an algorithm. In the supermarket context, blablablabla  (instead of Data flow diagram) -> state the purpose, link to context  4) How are inputs collected and are there any output reports like receipts, so that you know how to code the system.  5) Front end system (online form/interface)  6) Planning the actual low level algorithm, how the processes will actually work before coding the solution out.  7) Think of testing plans, if you’re doing test-driven development (think of testing before even start coding)  ~~4) Budget~~ (NOT RECOMMENDED BUT ACCEPTABLE: You can’t code if you got no money (you getting employed for this) or implement new features) |

The work to implement new uses of the customer data needs to be managed. The   
 following Program Evaluation and Review Technique (PERT) chart is used as a   
 management tool.

Time is measured in weeks.



**(d)** Each activity indicated by a dashed line on the PERT chart is a dummy activity.

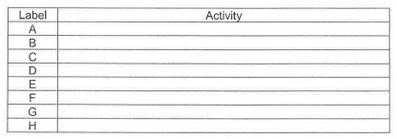
**(i)** Explain the nature and purpose of a dummy activity. [2]

|  |
| --- |
| Dummy activities are imaginary activities without any time duration to demonstrate relationships/dependencies between activities that would otherwise be difficult to show with simple arrow linkages. (REMEMBER TO PROVIDE AN EXAMPLE FOR 2M QUESTION)  For example, new hardware acquired can be used to host the database, yet is independent of the implementation/creation/coding of the database itself. Hence, a dummy activity can be used to draw a relationship/dependency between these two activities that would otherwise be on distinct paths in a PERT chart. |

**(ii)** Each of the following activities matches one of the labels AH on the chart.

* write user documentation
* train users
* write code
* convert files
* test code testing algorithm, white box, black box
* end-user testing
* test system testing among parts of the system
* install new hardware

Copy and complete the following table,



[4]

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Label | Activity | | A |  | | B |  | | C |  | | D |  | | E |  | | F |  | | G |  | | H |  | |

**(iii)** Explain the significance of the dummy activity that leads into event . [3]

|  |
| --- |
|  |

**(e)** From the PERT chart:

**(i)** State the critical path. [1]

|  |
| --- |
|  |

**(ii)** State the minimum time in which the new software can be developed and   
 implemented. [1]

|  |
| --- |
|  |

**(iii)** The chart omits an activity: **write technical documentation**. State a starting point   
 and a finishing point for this activity. Justify your choices. [4]

Management staff can already access the company network remotely for other software applications. Management are to be given the facility to access, and interact with, the customer data Via the company LAN. However, a decision is made not to allow access to the customer data remotely for this updated system.

**(f)** Describe **two** methods which can be used to ensure that there is no remote access to   
 customer data by management staff. [4]

|  |
| --- |
|  |

ln the new system, customers will have access to information through a web browser. Each customer will be able to see some information about their purchase history.

**(g)** Explain what software needs to be developed to provide this customer facility. [5]

|  |
| --- |
|  |

**(h)** One of the software developers has the task of ensuring that social issues are   
 considered.

This developer has to document these issues.

Describe **two** issues that might be in the document with regard to customers accessing   
 their data. [4]

|  |
| --- |
|  |

**2013**

**1** A dental practice currently uses a computer system to store details of its patients, staff   
 and appointments in separate files.

The practice manager and the receptionist have their own computers for accessing and   
 updating the files.

The system produces a small number of reports.

An updated system is to be produced by a software company. The updated system will   
 use a database. In the updated system the dentists will be given a hand-held device to   
 use in their rooms for accessing and updating the patient records. The new system will   
 also be capable of producing additional reports.

The software company has software engineers who have expert skills in specific areas   
 of software development. A number of the engineers will be involved in the development   
 of the updated system.

**(a)** Describe and justify three methods which can be used to determine what further   
 reports are required from the updated computer system. [6]

|  |
| --- |
|  |

**(b)** The work to update the system is partly managed by the following Program   
 Evaluation and Review Technique (PERT) chart.

A - investigation

B - analysis

C - design of database

D - design of reports

E - design of screen displays for dentists

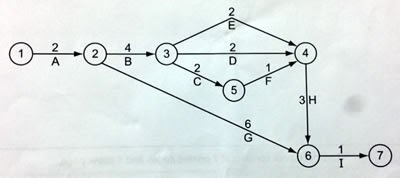
F - transfer of data from files into database

G - documentation produced

H - acceptance testing

I - hand over to customer

Time is measured in weeks.



**(i)** State the critical path. [1]

|  |
| --- |
| ABCFHI |

**(ii)** State the minimum time in which the updated system could be operational. [1]

|  |
| --- |
| 13 weeks. |

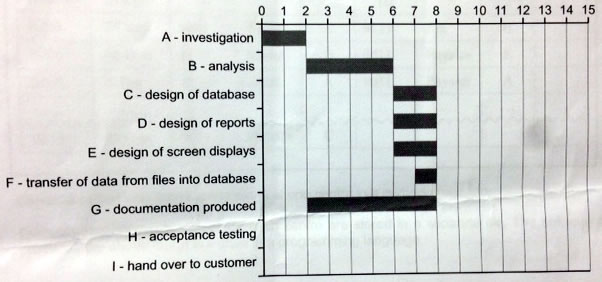
**(iii)** For activity E state the

* earliest Start time
* earliest Finish time
* latest Start time
* latest Finish time

[4]

|  |
| --- |
| Earliest Start time: Week 6  Earliest Finish time: Week 8  Latest Start time: Week 7  Latest Finish time: Week 9 |

**(c)**



The Gantt chart above is based on the information in part **(b)**. The timing of two   
 activities is missing and also the timing of one of the activities shown is incorrect.

Draw a sketch of the Gantt chart to show the correct version. [4]

|  |
| --- |
|  |

**(d)** Explain how the Gantt chart can help with the work that the software engineers have   
 to carry out. [2]

|  |
| --- |
|  |

**(e)** A small team is put together to consider security aspects of the updated system.

**(i)** Identify **two** possible members of the team and justify your choice. [4]

|  |
| --- |
|  |

The team have to produce a report to which they all make a contribution. The report   
 is stored on a network. Each member of the team has access to allow them to add   
 their contribution.

**(ii)** Give **two** examples of unethical behaviour by a team member. [2]

|  |
| --- |
|  |

**(f)** Name and describe **two** types of documentation produced for this project. [6]

|  |
| --- |
| End-User documentation   * for actual users of system to learn about features and how to use them * minimum/recommended hardware and software system requirements (operating system, version, processor, amount of RAM and hard disk space, etc.) * installation guide + step by step guide of how to perform a task or use a feature * frequently asked questions (FAQ) for common troubleshooting problems and solutions * support contact information, safety instructions, warranty information   Technical documentation   * for developers to document technical requirements and features of system * system objectives and scope * input and output/report specifications * data storage/database specification * modules/processes and algorithms * user interfaces and application programming interfaces (APIs) * testing * implementation/deployment * bugs report and known issues |

The hand-held devices the dentists use in their practice rooms will be networked. Both   
 client-side scripting and server-side scripting will be used in the new software which is   
 produced. An intranet with a web server will be created. Web browsers will be used on   
 the hand-held devices.

**(g)** Describe three possible uses of the device. [6]

|  |
| --- |
|  |

**(h)** For each scripting method, client-side scripting and server-side scripting, give an   
 appropriate example. Justify your response. [4]

|  |
| --- |
|  |

**2013**

**5** Bank customers are allowed to withdraw money from their accounts at an ATM. They   
 cannot withdraw more than the current balance in their account. There is a daily limit on   
 the amount that can be withdrawn. In some circumstances a charge is made for the   
 transaction.

The rules are:

* the transaction is rejected if the withdrawal amount requested is greater than the current balance
* the transaction is rejected if the withdrawal amount exceeds the daily limit
* if the current balance before the transaction is carried out is less than 50 dollars then any successful transaction incurs a fixed charge

**(a)** Create a decision table showing all the possible conditions and actions. [4]

|  |
| --- |
|  |

**(b)** Simplify your decision table by removing redundancies. [4]

|  |
| --- |
|  |

**(c)** Using your answer in (b) write a function using pseudocode. The function returns:

* -1 to indicate a rejection;
* 0 for a charge-free successful transaction;
* the charge for a chargeable successful transaction.

[5]

|  |
| --- |
|  |

**(d)** State two ways in which your answer in (c) demonstrates clarity of code. [2]

|  |
| --- |
|  |

**SP9597**

**1** In a school each teacher is expected to keep paper records on each student that they   
 teach. The teacher mark book contains data on student results in examinations, tests   
 and homework assignments. Examination data for each student is extracted from the   
 school examination summary results sheets.

In addition other records are also kept on one-to-one interviews between the teacher   
 and the student (using an interview form) where recent performance and previous   
 targets are discussed and new targets are set.

The school principal requires a regular report on each student's performance and new   
 targets set. The teacher has to produce this report from information on the interview   
 forms.

The school wishes to replace this manual system with a computerised system. A system   
 developer is employed to carry out the task. The first task assigned to the system   
 developer is to write a project proposal.

**(a)** One section of the project proposal is the Problem Statement which lists the   
 problems in the current system. Write the Problem Statement. [6]

|  |
| --- |
| Problem statement might include the following issues:   * storing teacher records of students - volume, accessibility * organization of data in teacher markx book - layout, volume, accessibility * accessing teacher records in mark book - by whom * manual extraction of student exam data - time issues * making an accurate copy of the relevant student data - transcription errors * storage of interview forms - volume, accessibility * accurate recording of interviews - accessibility, volume * storage of recent performance and previous targets - accessibility, volume * recording of new targets - accessibility by teacher and student * production of report for principal - manual summary time, volume issues   1 mark per point to a maximum of 6 |

**(b)** The proposal is accepted and the system developer produces the following Program   
 Evaluation and Review Technique (PERT) chart:

A - analysis of the solution

B - design of the solution

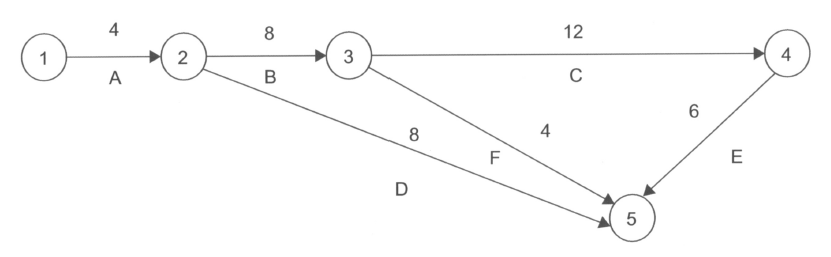
C - development of the solution

D - documentation of the solution

E - implementation of the solution

F - testing of the solution

Time is measured in weeks.



**(i)** State the critical path. [1]

|  |
| --- |
| * ABCE |

**(ii)** State the minimum time in which the project could be completed. [1]

|  |
| --- |
| * 30 weeks |

**(iii)** Explain dependent stages and concurrent stages. For each type of stage give an   
 example from this chart. [4]

|  |
| --- |
| * dependent stages are stages where one stage cannot be started until another one has been completed * e.g. design stage cannot start until analysis stage has been completed * concurrent stages are stages which can happen at the same time * e.g. documentation stage can happen whilst the design stage is happening |

**(c)** A decision is made that the PERT chart should show more detail with regard to   
 testing. It is proposed that stage F (testing) should be removed from the chart and   
 three new stages added:

L - black box testing - 2 weeks

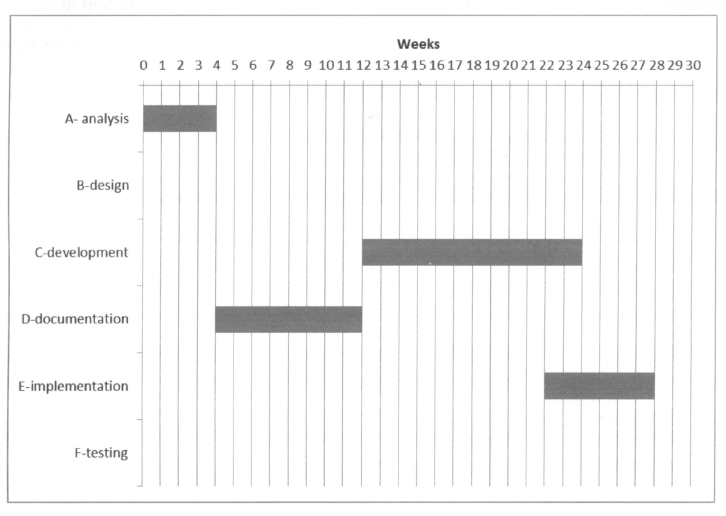
M - white box testing - 2 weeks

N - beta testing - 3 weeks

Redraw the PERT chart to show the effect of these changes. [4]

|  |
| --- |
| * stages L and M start at point 3 * and finish at point 5 * stage N between points 4 and 5 * stage F removed |

**(d)**



The Gantt chart above is based on the information in part (b). There are two activities   
 missing and also one of the activities shown is incorrect. Draw a sketch of the Gantt   
 chart to show the correct version. [4]

|  |
| --- |
| * B drawn correctly * F drawn correctly * identification of E as the correct activity * E drawn correctly |

**(e)** As a result of the analysis carried out a dataflow diagram (DFD) is used to   
 summarise the current system. This DFD is used in the initial stages of the design.

**(i)** Draw a DFD for the system described above. [6]

|  |
| --- |
| * two correct external entities * two correct processes * two correct data stores * each correctly identified data flow to a maximum of 3 |

**(ii)** By using examples from your DFD explain how the diagram helps to specify three   
 important components of the current system. [6]

|  |
| --- |
| * files on DFD - identify tables/files needed in database/file based solution * e.g. teacher mark book / file * processes on DFD - identify possible macros/code/queries/menu options * e.g. find student result * data flows on DFD - identify possible reports * e.g. current student performance report * data flows on DFD - identify possible screen forms * e.g. conduct and record interview   2 marks per component to a max of 6 |

**(f)** The design for the new system includes the provision of a network of computers in   
 the school with a central file server. Each teacher will have access to a computer to   
 retrieve and update student data held on the central file server. Some support staff   
 are allowed to access the data but not change it. In addition the system has an   
 Internet link which allows teachers to access the system from outside the school.

Describe three ways in which the security of this system can be maintained. [6]

|  |
| --- |
| * only staff allowed to access system * - allocated username and password * some staff are only allowed to view data * - application of access rights * prevent unauthorised external access to school system * - use of firewall * prevent malware, viruses * - use of anti-virus software / restrict use of exchangeable storage devices * ensure data is unreadable if successful hacking occurs * - encryption of data   max 2 marks per issue, max 3 issues |

**(g)** It is proposed that the system is also made available to students and their parents.   
 Both students and their parents would be allowed to add data to the system.

Give an example of what parents and students could be allowed to do and state the   
 ethical issue that may arise out of this. [2]

|  |
| --- |
| * add comments * check on honesty / objectivity of comments added |

**2012P1**

**2** A manual system for producing school student reports works in the following manner:

* a subject report is completed for each subject that a student takes by the single teacher teaching that subject;
* to help the subject teacher, initially a blank report form is issued to the student for the student to add their details: name, teacher and class;
* the subject report is completed by the teacher with appropriate comments;
* all subject reports for the student are passed to the student's tutor
* the tutor puts all the subject reports together to form the student's report folder;
* the tutor adds a tutor's report including attendance data supplied by the school administration attendance records;
* each student's report folder is copied;
* the copy is filed in the report storage facility for the school;
* the report folder is sent to the student's parents.

The school has decided to replace this manual system with a computerised system. A   
 systems developer is employed to carry out the task.

**(a)** Describe the steps that have to be taken before the design of the solution can be started. [6]

|  |
| --- |
|  |

**(b)** Draw a data flow diagram for the manual system. [8]

|  |
| --- |
|  |

**(c)** Explain how a data flow diagram helps in determining input and output requirements. Use   
 examples from your data flow diagram to illustrate your answer. [4]

|  |
| --- |
|  |

**(d)** Explain top-down analysis and why it helps in the solution of complex problems. [4]

|  |
| --- |
|  |

**2012P1**

**4** A supermarket uses a computerised stock control system. The system has point-of-sale   
 terminals staffed by checkout operators.

**(a)** Describe a possible computerised stock control system for use in the supermarket. In your   
 description explain how sales data are captured and how stock levels can be maintained.  
 [6]

|  |
| --- |
|  |

**(b)** State **two** disadvantages to the supermarket in having checkout operators using   
 point-of-sale terminals. [2]

|  |
| --- |
|  |

The supermarket decides to extend their system so that customers have the facility to check   
 out goods themselves.

**(c)** Describe **two** different methods that could be used to implement this new facility. [4]

|  |
| --- |
|  |

**(d)** Describe **one** economic advantage to the supermarket of these new methods. [2]

|  |
| --- |
|  |

**(e)** State **one** disadvantage of these new methods. Describe how this disadvantage might be   
 overcome. [2]

|  |
| --- |
|  |

**(f)** Describe steps that might be taken to limit the effects on the supermarket of a failure of the   
 computer system. [4]

|  |
| --- |
|  |

**2011P1**

**5** An electricity supply company places a meter in each domestic customer’s residence. The meter is   
 read every six months by a meter reader who is an employee of the electricity supply company.   
 Three months after each reading an estimate is made of the electricity consumed and is added to the   
 customer account details. Customer account details are held on a computer system.

**(a)** Name **two** items of data that need to be recorded at each meter reading. [2]

|  |
| --- |
|  |

**(b)** Describe **two** ways in which this data could be captured by the meter reader so that there is no need   
 to key the data in when it is entered into the computer system. [4]

|  |
| --- |
|  |

**(c)** Describe **two** reports output from this computer system. [4]

|  |
| --- |
|  |

**(d)** The supply company decides to implement an automatic meter reading system.

Explain how such an automatic system could work. [2]

|  |
| --- |
|  |

**(e)** State

**(i)** **two** benefits of this new system for customers, [2]

|  |
| --- |
|  |

**(ii)** **two** benefits of this new system for the company. [2]

|  |
| --- |
|  |

**(f)** Describe the benefits to society of automated utility systems. [3]

|  |
| --- |
|  |

**2011P1**

**7** Systems analysts have skills in developing computer systems. Their customers are unlikely to have   
 these skills but will have in-depth knowledge of their business. A systems analyst hired by a business   
 will have little knowledge of that business initially.

**(a)** Explain how the managers of the business can indicate to the systems analyst what they require. [2]

|  |
| --- |
|  |

**(b)** There are a number of methods available to the systems analyst to gather more information on the   
 current system used by the business.

Give **three** methods of information collection which the analyst can use, justifying the use of each.   
 [6]

|  |
| --- |
|  |

**(c)** If a programming solution is chosen describe what decisions the systems analyst has to take before   
 coding starts. [6]

|  |
| --- |
|  |

**(d)** The developed code should meet the customer requirements.

Explain how this can be shown to the customer’s satisfaction. [2]

|  |
| --- |
|  |

**2010P1**

**3** A car insurance company allows customers to insure their car either by using the internet or by using   
 the telephone to talk to a sales representative. Both methods depend upon the company having a   
 reliable computer system.

**(a)** State two problems that could result in system failure, giving a solution for each. [4]

|  |
| --- |
|  |

**(b)** Explain why customers may be reluctant to use the internet option. [3]

|  |
| --- |
|  |

**(c)** Explain what is meant by data privacy. [2]

|  |
| --- |
|  |

**(d)** State two reasons why the company needs to ensure data privacy. [2]

|  |
| --- |
|  |

**(e)** Describe how the car registration number can be captured accurately when:

**(i)** customers use the internet,

|  |
| --- |
|  |

**(ii)** customers talk to a sales representative on the telephone. [4]

|  |
| --- |
|  |

The rules that are used when deciding whether to offer insurance to customers and whether to offer

discounts are as follows:

* If the customer has been refused insurance by another company and their car is over 10 years old then insurance is refused.
* If the customer has been refused insurance by another company and their car is not more than 10 years old then insurance without any discount is available.
* If the customer has not been refused insurance by another company and their car is over 10 years old then insurance without any discount is available.
* If the customer has not been refused insurance by another company and their car is less than 10 years old and they have made not more than three claims previously then insurance with a discount is available.

**(f)** Create a decision table showing all the possible outcomes and results. [4]

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Conditions** |  |  |  |  |  |  |  |  | | Refused insurance by another company | Y | Y | Y | Y | N | N | N | N | | Car is over 10 years old | Y | Y | N | N | Y | Y | N | N | | Have made more than 3 claims | Y | N | Y | N | Y | N | Y | N | | **Actions** |  |  |  |  |  |  |  |  | | Insurance refused | X | X |  |  |  |  |  |  | | Insurance without discount |  |  | X | X | X | X | X |  | | Insurance with discount |  |  |  |  |  |  |  | X | |

**(g)** Simplify your decision table by removing redundancies. [5]

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **Conditions** |  |  |  |  |  | | Refused insurance by another company | Y | Y | N | N | N | | Car is over 10 years | Y | N | Y | N | N | | Have made more than 3 claims | - | - | - | Y | N | | **Actions** |  |  |  |  |  | | Insurance refused | X |  |  |  |  | | Insurance without discount |  | X | X | X |  | | Insurance with discount |  |  |  |  | X | |

**2010P1**

**5** A systems analyst is developing a new computerised admission system for a college.

* Each new student sends a completed form that has their name, date of birth and the courses that they wish to enrol on.
* The date of birth is checked to see whether the student is of the correct age range for admission to the college.
* If the student is too young or too old a standard rejection letter is produced.
* If the student is of the correct age their personal details are added to the student file.
* If the student is of the correct age then each of the courses that the student has identified are checked on the course file to see whether they are full or not.
* If there is room on a course then the student name is added to the appropriate course record on the course file.
* A standard letter is produced with details of which course(s) the student has been enrolled on.

**(a)** The systems analyst was given a detailed problem definition initially. Why is this important? [3]

|  |
| --- |
| precise problem definition (objectives, requirements, priorities)   * accurately understanding system requirements * knowing the scope i.e. what is included and what to ignore * knowing when a successful solution has been reached * allocate scarce resources for higher priority items |

**(b)** Draw a data flow diagram (DFD) for the current system. [6]

|  |
| --- |
| please remember to use only the 4 correct types of symbols  rectangle - external entity  ellipse - process  open rectangle with double vertical lines on left hand side - data store  data flow must be arrowed (shows read from / write to relationship) and must be labelled |

**(c)** Explain, with examples, how the DFD would help develop a database solution for this system. [6]

|  |
| --- |
| * data stores - show tables needed * data flow from/to processes - shows attributes/fields in database table structure/design * data flow from/to external entities - shows input interfaces with data requirements and output screens/reports to users * processes - contain computation which may be read from or written into database tables |

**(d)** Give one part of the database design that is not possible from the DFD. [2]

|  |
| --- |
| * does not explicitly identify primary key, secondary index, foreign key, composite key makeup * does not show optimal/normalized database design * does not enforce referential constraints eg duplicate or null/empty primary key |

**2010P2**

**6** The new database system was designed by a systems analyst.

**(a)** The first task of the analyst is to collect information about the old system and the requirements of   
 the new system.

State **three** methods of collecting information and justify their use in this application. [6]

|  |
| --- |
| * examine current system documents - identify data requirements and processes, can be performed off-site without disturbing existing operations * observe current work procedures - existing procedures need to be computerized, ensure new system with integrate well with other aspects of business * conduct surveys and questionnaires - get quantitative and qualitative input from large group of users, anonymity may lead to more truthful answers * interview users across different levels of organization - to clarify objectives, requirements and other issues from other investigation methods, different perspectives of system |

**(b)** When the problem has been fully analysed, the analyst decides that a modular solution should be

created.

Explain why a modular solution is chosen in preference to a single program. [4]

|  |
| --- |
| * modular solution allows a large complex solution to be decomposed into smaller and more managable parts/modules which can be solved in a more systematic/structured manner * these modules can be independently worked on and tested by different people/teams hence benefitting from productivity gains by division of labour * bugs/errors can also be more easily isolated and debugged/rectified * testing can also commence earlier to test completed modules and incomplete modules can be implemented using dummy stubs (eg print statements) |

**(c)** When the solution is complete the system must undergo thorough testing.

**(i)** Explain the difference between black box and white box testing. [2]

|  |
| --- |
| Black box testing   * test functionality by providing input-output specification (what) * no explicit knowledge of implementation details, only interface (input and output) * can be performed by system testers (non-technical)   White box testing   * ensure all possible control paths are exercised/tested and produce expected results (how) * requires explicit knowledge of code within module * usually performed by programmer/developer |

**(ii)** Each module has been fully tested and works. Explain why it is possible that when these   
 modules are combined they do not work properly. [3]

|  |
| --- |
| * module may have passed unit testing but not integration and system testing * interfaces/dependencies between modules are not tested properly/fully eg output format from one module is incompatible with input to receiving module such as data type or order of parameters * algorithm may not be efficient enough to cater to volume data, hence slow processing leading to illusion that program fails to run |

**(iii)** One module is designed to add the monthly payment (M) for an apartment to a cumulative   
 total (T).

State **two** sets of test data for this example, explaining why they should be used. [4]

|  |
| --- |
| * ensure both monthly payment (M) eg M = 123.50 and total (T) eg T = 0.00 are declared as real/float/double data type variables, so that no truncation errors occur * alternatively use integer to store monetary values correct to nearest cents eg M = 12350 and T = 0, as some real numbers cannot be represented using a finite number of bits allocated for numeric data storage, and hence will introduce truncation errors * ensure T is declared using double precision data type to prevent overflow arithmetic errors |

**2009P1**

**6** A bank is worried about computer crime.

   One major concern is on-line access to customer accounts.

**(a)** Give **two** ways that a customer's on-line access details could be discovered by a criminal. [4]

|  |
| --- |
| * Keyloggers that log the keystrokes of the customer as it logs into the bank. * Directly hacking into the bank’s servers * Spoofing: Listening in on the network to obtain customer’s login details. * Phishing: Fake an official website to collect customer’s restricted information. |

**(b)** Describe **one** other way in which a bank customer could be the victim of computer crime. [2]

|  |
| --- |
| * Virus and trojans from sites posing to be the official bank site. * ATM card readers that steal your ATM information |

**(c)** Explain why a customer using a bank card for on-line shopping may be more of a security issue   
 than customers using cards in shops. [3]

|  |
| --- |
| * difficult to verify information beyond credit card details online eg user identity behind terminal * customers using cards in shops may be required to give a signature * cashier can perform a facial match of the customer against the photo on the bank card * shops can also have security cameras to capture customer photo in case fraud is suspected |

   The bank decides that another level of security is required for on-line customers.

   Each customer is given a bank card reader.

   If they wish to do an on-line transaction:

* they have to put their bank card in the card reader
* input their PIN
* the card reader produces a unique number which has to be entered into a computer if the transaction is to be carried out successfully

   There is no connection between the card reader and the computer.

**(d)** Explain how this new system improves security. [3]

|  |
| --- |
| * card reader cannot be accessed via the Internet thus it cannot be hacked unless the hacker gains physical access to the reader * PIN of the customer is required * card must be physically in the user’s possession. * this improves security as the hacker must obtain the card, reader and get hold of the customer’s PIN in order to complete the transaction |

**2009P2**

**6 (a)** State what is meant by utility software. [1]

|  |
| --- |
| * program that performs automated or productivity tasks on top of the operating system * eg useful functions such as file compression, word processing, images editing, etc. |

**(b)** Explain what is meant by each of the following types of utility software. Give an example of how it

would be used by the staff of the company.

**(i)** Compression software  [2]

|  |
| --- |
| * compact/reduce the size of files without losing essential data * to save storage space or facilitate faster transfer * files can be compressed by removing non-essential white space for text files or only storing incremental differences within graphic/video files * can also consolidate many files into a compressed folder/archive for easy transfer * company can use compression software to reduce the size of the documents on the server so that employees can easily and quickly download the files for their work |

**(ii)** Virus checker [2]

|  |
| --- |
| * program that runs as a background process to constantly monitor user or system activity for malware lurking within files or data * once detected and verified with virus definition file, will alert user with recommended action to remove threat and clean system * if unable to clean or if virus is suspected but does not correspond exactly with virus definition file, will rename file with non-executable extension and move file to quarantine folder and inform user about the action * company will need to install virus checker to constantly scan working files and activities on the network for malware, or perform scheduled updates and scans * when malicious activity occur, virus checker will recommend actions to be taken and even notify the IT department to check on the employee’s machine infected |

**(iii)** Hardware driver [2]

|  |
| --- |
| * program that contains instructions required to communicate with and control a peripheral, to interpret and send/receive instructions to and from it * staff can download and install hardware driver for network printer to send print jobs to a shared printer * or company can use hardware driver for a barcode reader to read barcodes on stock items for inventory tracking and management |

**(c)** The customer file is very important to the company. Backups and archives are taken regularly.

Explain the difference between backups and archives. [4]

|  |
| --- |
| Backups   * duplicate copy of current working files * to quickly restore from backup to a useful working state in the event the original file is corrupted or lost (eg hard disk crash) * usually stored on an equivalent access medium eg another hard disk accessible on the same machine/server or remotely to another location via network transfer * frequency of backup dependent on activity level of original file => the more frequent the main copy is changed, the more frequent backup should be performed   Archives   * copy of original files made at longer regular intervals (eg yearly) and stored away for historical purpose, future audit or legal requirements or ad-hoc reference eg past year customer records * often compressed to minimise storage space due to infrequent access * usually stored separate from current working main copy to slower access media or offline |

**2008P1**

**3.** A food store does not use a computer for its stock control. Each item of stock that it sells has a price

label attached. These labels are used at the checkout to key in each price manually. After all the   
 items have been entered the total is calculated and printed.

**(a)** State one advantage and one disadvantage to the customer of this non-computerised solution.   [2]

|  |
| --- |
| Advantages:   * Price haggling can be done * Convenient for the consumer to know the price of each product. * Manual data entry error [allows you to display your integrity]   Disadvantages:   * Manual data entry error * Slower checkout times |

**(b)** State one advantage and one disadvantage to the management of the food store of this   
 non-computerised system.         [2]

|  |
| --- |
| Advantages:   * No need to spend on expensive computerised equipment * No need to train the staff * No need to train yourself   Disadvantages:   * More manual error * More difficult to maintain accurate inventory * Lose long-term competitiveness * Slower checkout times * Finger more pain |

   The store decides that it is necessary to computerise their stock control.

**(c)** Describe a possible computerised stock control system for the store.                         [4]

|  |
| --- |
| * Use barcode scanner to scan barcodes on labels for product ID * Store all product details in a random access file ordered by product ID * When barcode is read display details * Update stock quantity when product is considered bought * Trigger stock re-order when stock level goes below desired level |

**(d)** Explain two economic advantages for the store of implementing the computerised stock control   
 system.                                                                                                 [4]

|  |
| --- |
| * Faster turnaround time => Serve more customers => Better customer experience => More customers => ??? => PROFIT * Higher productivity level per worker => Better able to serve the needs of customers at the minimum of staff => lower cost of production => ??? => PROFIT |

**2008P1**

**6.** A computer system is to be developed from a current manual system. The problem is to be solved by   
 writing programs. A number of stages are necessary before programming can commence.

**(a)** Why is it necessary to define the problem as precisely as possible?                         [2]

|  |
| --- |
| precise problem definition (objectives, requirements, priorities)   * accurately understanding system requirements * knowing the scope i.e. what is included and what to ignore * knowing when a successful solution has been reached * allocate scarce resources for higher priority items |

**(b)** State three methods that can be used to collect information on the current system. Give an   
 advantage for each of the stated methods.                                                       [6]

|  |
| --- |
| information collection methods (please give advantage applied to context)   * study current system documents * observe current work procedures * conduct surveys and questionaires * interview key stakeholders across different levels of organization |

**(c)** Describe two different types of diagram that may be produced at the analysis stage.     [4]

|  |
| --- |
| * data flow diagram - shows data/information flow across entities (input, output), processes and data stores within the system, does not specify types of data stores * system flow diagram - shows data/information flow across various **types** of input (manual, system), output (screen, document) and storage devices (tape, disk) * block diagram or structure/organization chart - shows how current system is organized in a modular view   note: program flowchart not acceptable. why? because analysis stage is on current/old system, whilst program flowcharting is for proposed/new system |

**(d)** Explain top-down analysis and why it helps in the solution of complex problems.             [4]

|  |
| --- |
| note: analysis, not design   * breaks down complex problems into smaller, simpler and more manageable sub-problems * different people/teams can work on and understand different sub-problems * different problems can be studied independently/concurrently at different pace * easier to understand smaller problems to apply logical methodology to produce analysis diagrams as output |

**2008P2**

**3.** Customers who book a lane using the website are expected to pay for the booking over the Internet.

**(a)** Non-members must pay for bookings immediately.

**(i)** Describe the problems that may arise with this transaction for both the customer and the   
 company.                                                                                             [4]

|  |
| --- |
| Company:   * Customers find it inconvenient to pay for bookings upfront over the Internet; company loses revenue. * Unable to attract teenager crowd since they cannot pay online * Expensive to obtain and properly maintain the structures and dependencies required for safe credit card transactions   Customer:   * Customers that cancel booking at the last minute may still be charged for the booking since it is hard for company to cancel credit card charges. * May not have means to pay online especially teenagers |

**(ii)** Describe measures which can be taken to overcome these problems.                     [4]

|  |
| --- |
| * Make everyone create an account. * Allow non-members to pay in cash at the venue before the booking time. * Allow non-members to transfer the cash to the company’s current account and then quote the transaction number in the booking. * Have hotline to credit card companies that are accepted. |

**(b)** Members have an account with the company so that charges may be billed directly to their   
 account.

**(i)** Explain why some members may not want to allow the company to store such details. [3]

|  |
| --- |
| * Personal information may be exploited if the company servers are hacked * Privacy of members are compromised. * Cause unsolicited advertising |

**(ii)** Describe the measures which the company can try to overcome any concerns.       [3]

|  |
| --- |
| * Privacy clause on company website * Delete all sensitive data about members * Ensure that all data is stored and transmitted in heavily encrypted form * Prepare lawyers |

**2007P1**

4. A systems analyst has been employed to develop a computerised system for a school library which   
 is currently operating manually.

* Each book has a unique book card.
* Each borrower has a number of tickets.
* When a book is loaned from the library the book card and one of the borrower's tickets are placed together and stored in a tray.
* When the book is returned the book card and the ticket are located in the tray.
* A check is made to see whether the book is on the overdue list.
* The card is placed back in the book and the ticket is returned to the borrower for future use.

**(a)** Give three advantages of using a data flow diagram to describe how the current system works. [3]

|  |
| --- |
| * identify entities, processes, data/information flow and data stores within the system * shows flow of data throughout different parts of system, shows input and output * helps in analysis of current system information requirements * high level visual overview of current system without technical jargon which can be used for communication/verification with users, easier to understand |

**(b)** Draw a data flow diagram for the current system. [6]

|  |
| --- |
|  |

**(c)** State three methods the system analyst could use to collect information on the current system.   
 Give an advantage for each of the stated methods. [6]

|  |
| --- |
| * study current system documentation (forms, reports, etc.) - identify data items requirements, objectives, scope, input, output * observe current work processes - understand procedures to be computerized, integration with existing/other work processes * conduct surveys and questionaires - obtain quantitative and qualitative input/feedback from large number of users, provide anonymity for more truthful answers * interview users across different levels of organization - clarify objectives, requirements and other issues from other investigation methods, get different perspectives from different user segments |

The overdue list is processed in the following manner:

* If a book is overdue then a reminder letter would normally be sent.
* However, if the book is more than five days overdue two further checks are made to see whether the reminder should be replaced by a warning letter:
  + If the student has had a previous warning letter the student will not only receive the   
     warning letter but, in addition, a copy will be sent to the parents of the student.
  + If the student has more than four books overdue, but no previous warning letter, the    
     reminder letter is replaced by a warning letter.

(d) Create a decision table showing all the possible outcomes and results.   [4]

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Conditions** |  |  |  |  |  |  |  |  | | More than 5 days overdue | Y | Y | Y | Y | N | N | N | N | | Has previous warning letter | Y | Y | N | N | Y | Y | N | N | | More than 4 books overdue | Y | N | Y | N | Y | N | Y | N | | **Actions** |  |  |  |  |  |  |  |  | | Send reminder letter |  |  |  | X | X | X | X | X | | Send warning letter to student | X | X | X |  |  |  |  |  | | Send warning letter to parents | X | X |  |  |  |  |  |  | |

(e) Simplify your decision table by removing redundancies. [4]

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | | **Conditions** |  |  |  | | More than 5 days overdue | Y | Y | N | | Has previous warning letter | Y | N | - | | More than 4 books overdue | - | Y | - | | **Actions** |  |  |  | | Send reminder letter |  |  | X | | Send warning letter to student | X | X |  | | Send warning letter to parents | X |  |  | |

**SP9754P1**

**1** A systems analyst has been employed to develop a computerised system for   
 order processing in a retail business.

**(a)** State three methods the analyst could use to collect information on the current   
 manual system. Give an advantage for each of the stated methods.                       [6]

|  |
| --- |
| * examine current system documents - identify data requirements and processes, can be performed off-site without disturbing existing operations * observe current work procedures - existing procedures need to be computerized, ensure new system with integrate well with other aspects of business * conduct surveys and questionnaires - get quantitative and qualitative input from large group of users, anonymity may lead to more truthful answers * interview users across different levels of organization - to clarify objectives, requirements and other issues from other investigation methods, different perspectives of system |

**(b)** The system analyst, when documenting the system, can use a number of   
 diagrammatic methods to support what is written. Give three examples of diagrams   
 that might be found in the technical documentation and explain why they are   
 included.   [6]

|  |
| --- |
| * Unified Modelling Language (UML) class diagrams - specify attributes/data and operations/methods of entities/classes, show relationship (eg inheritance) between classes for code reuse * Entity-Relationship (ER) diagrams - shows relationships/dependencies (eg one-to-one, one-to-many, many-to-many) between entities/database tables * Data Flow Diagrams (DFD) - shows the flow of data/information within different entities and processes and data stores in the system * System Flow Diagram/Chart - shows high-level organization of system at physical/resource level * Program flowchart - provides implementation-independent sequence of logical steps to accomplish a task * Block diagram or structure/organization chart - shows top-down modular breakdown of system design * Decision tree - shows possible actions based on a set of conditions/criteria |