

1 A software developing company needs software to calculate how much each employee should be paid.

[3]

Ans

1 mark per row

		Rules							
Conditions	Public holiday	Y	Y	Y	Y	N	N	N	N
	Hours >= 160	Y	Y	N	N	Y	Y	N	N
	Pension	Y	N	Y	N	Y	N	Y	N
Actions	3% bonus payment	X	X	X	X				
	5% bonus payment	X	X			X	X		
	4% pension payment	X		X		X		X	

(1)

(1)

(1)

(b) Explain the changes in the decision table after removing redundancies.

[1]

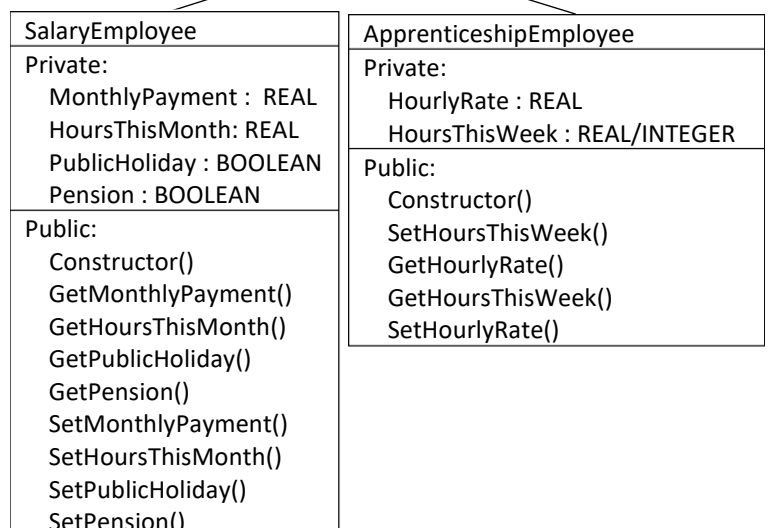
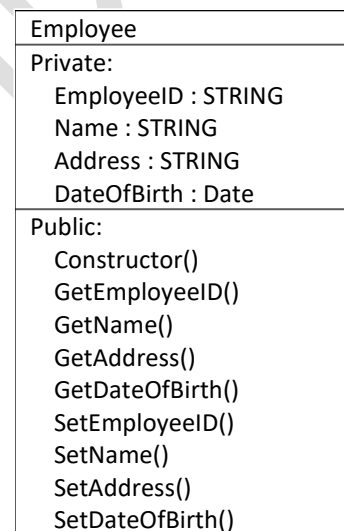
Ans: No pair of actions for a several combinations are the same. so no combinations can be combined, and no reducing the number of rules (1)

(c) The company has decided to implement a program for the software using object-oriented programming (OOP).

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

Ans:

- inheritance (arrow filled or unfilled) (1)
- 3 class names (1)
- Private attributes (2)
- Public methods (2)
- Accessors/Getters (2)
- Mutators/Setters (2)



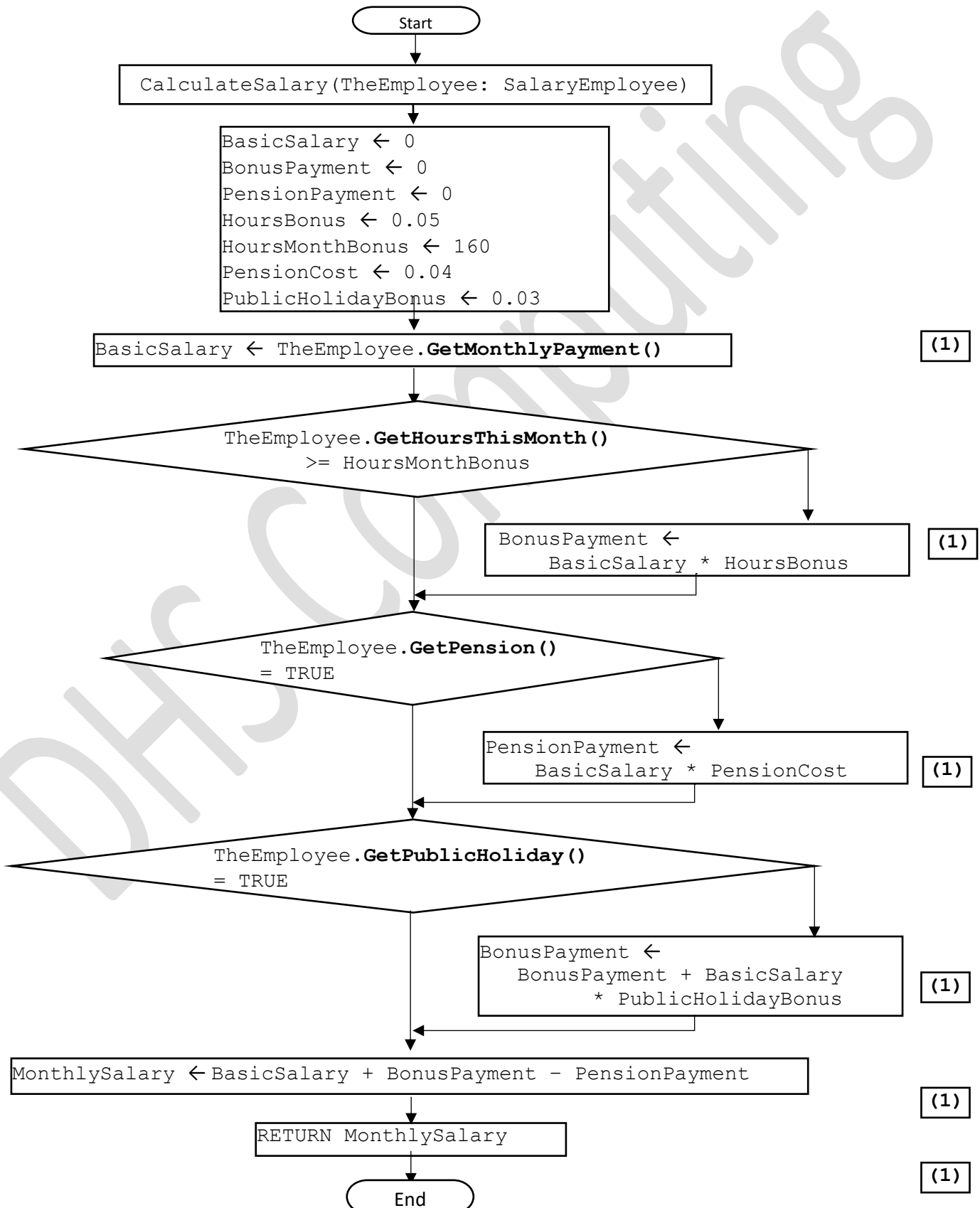
(d) A SalaryEmployee is paid a fixed monthly payment.

Copy and complete the flowchart below for the function CalculateSalary().

[6]

Ans: (1) mark per bullet point to max 6

- MonthlyPayment accessed from the parameter using Get method and store in BasicSalary
- Hours, Pension and PublicHoliday accessed from the parameter using Get methods
- Checking if Hours is ≥ 160 , calculating bonus payment ($\text{BasicSalary} * 0.05$)
- Checking if pension is true, calculation pension to pay ($\text{BasicSalary} * 0.04$)
- Checking if public holiday is true, calculation of bonus payment ($\text{BasicSalary} * 0.03$)
- Adding both bonus payments to basic salary and deducting pension from salary
- Returning the new salary



(e) (i) In OOP, state the purpose of polymorphism.

[1]

Ans: Allows objects or methods to exhibit different behaviors under different classes. (1)

(ii) What changes in design needs to be made if the objects of class

ApprenticeshipEmployee also use CalculateSalary() to calculate the weekly salary. [1]

Ans: CalculateSalary() must be a **public method** of ApprenticeshipEmployee (1)

(f) Discuss the advantages of using modular design in software development. [2]

Ans: Advantages including easier maintenance and debugging, enhanced reusability of components, improved collaboration, increased scalability, faster development cycles, and better fault isolation. (2)

(g) The software design may need to change from time to time by a team of programmers.

Explain how the team tracks the changes with a system. [2]

Ans: Version control is a system that tracks changes to files, typically code in software development, allowing teams to collaborate effectively and revert to previous versions when needed (2)

2 A company has a website. Users use the internet and the world wide web to access the website.

(a) Describe the difference between the internet and the world wide web. [1]

Ans: (1)

- ☐ The internet is the infrastructure
- ☐ The world wide web is a collection of web pages

(b) Data packets are transmitted across a network from one computer to another computer.
Describe the structure of a data packet. [1]

Ans: All **three** from: (1)

- ☐ A packet is split into **three** different sections
- ☐ ... the header
- ☐ ... the payload
- ☐ ... the trailer

(c) Explain what is meant by the term router and describe the function of a router in a computer network. [2]

Ans: A router is a device in a network which holds information about the addresses of computers in the network (or other networks) ... (1)

- Any one of the function of a router (1)
- ... and can direct data to the correct computer
 - ... and can direct data around network in (most) efficient way
 - ... and act as a gateway connecting to a larger network.

(d) Explain what is meant by circuit switching and packet switching in a computer network and give two advantages of packet switching over circuit switching. [4]

Ans: Circuit switching: (1)

- Path is set up between the sender and receiver
- All data follows the same path, in order
- Path cannot be used by any other data

Packet switching:

- Any 2 of for one mark total: (1)
- Data split into packets
 - Each packet may be transmitted by different routes
 - Packets may arrive out of order and are re-assembled

Packet switch preferred:

any 2 of for one mark each (1+1)

- Better security as it is very difficult to intercept
- Makes more efficient use of data lines as there is no waiting during gaps
- Less likely to be affected by network failure because multiple paths used

(e) A website of a company may collect *information of the customers*.

(i) State the meaning of *privacy* of data. [1]

Ans: (1) mark for: Either

- Ensuring data can only be accessed by / disclosed to authorised persons Or
- Ensuring data cannot be accessed by / disclosed to unauthorised persons

(ii) State the meaning of integrity of data. [1]

Ans: (1) mark for each bullet point (max 1)

- Ensuring the accuracy / completeness / consistency of data (during / after processing)
- Ensuring the data is up to date

(iii) Describe the following threats to a computer system.

[4]

Spyware
Phishing email

Ans: (1) mark for each bullet point.

Phishing email (**max 2**)

- The email pretends to be from an official body
- ... persuading individuals to disclose private information // by example such as bank details
- ... or requesting authentication by redirecting to an unofficial/unauthorised website // inviting a user to click a link

Spyware (**max 2**)

- Malware downloaded **without the user's knowledge**
- ... which secretly records the user's actions / keystrokes on the computer
- ... and sends logs of the actions to a third party

3 Details of errors generated in a program are stored in a stack.

(a) State which error will be the first retrieved from the stack.

[1]

Ans: The last one in // most recent

(b) The stack is implemented as a 1D array with the identifier `ErrorArray`.

- (i) The function, `AddItemToStack`, takes the next error, the array, and pointer as parameters. If the stack is full, the function returns `FALSE`; otherwise, it adds the error to the stack, changes the pointer's value and returns `TRUE`.

Copy and complete the following pseudocode for the function `AddItemToStack`. [4]

Ans:

1 mark for True and False in the correct place

1 for each other completed statement

```
FUNCTION AddItemToStack(BYREF ErrorArray : ARRAY[0:99] OF Error,  
    BYREF LastItem : INTEGER, BYVALUE Error1 : Error) RETURNS BOOLEAN  
    IF LastItem = 99 // ErrorArray.Length - 1  
        THEN  
            RETURN FALSE  
        ELSE  
            ErrorArray[LastItem + 1] ← Error1  
            LastItem ← LastItem + 1  
            RETURN TRUE  
        ENDIF  
    ENDFUNCTION
```

- (ii) Explain the reasons why `ErrorArray` and `LastItem` are passed by reference, but `Error1` is passed by value. [3]

Ans:

1 mark per bullet point to max 3

- The function needs to change the values in `ErrorArray` and/or `LastItem` in main/where called
- ... otherwise they would not be changed outside of the function // otherwise changes would only stay in the function
- `Error1`'s value does not change in the function // no changes to `Error1`'s value need reflecting where it was called / to the original
- `BYVALUE` stops the value being changed outside the function, but `BYREF` changes the value where called from

- (iii) The function `RemoveItem` takes the next error from the stack and returns it.
 If there are no errors in the stack, it returns the global object `NullError`.
 Copy and complete the pseudocode algorithm `RemoveItem`.

[3]

Ans:

- 1 mark for **both** return statements
 1 mark for each other completed statement

```
FUNCTION RemoveItem(ByRef ErrorArray : ARRAY[0:99] OF Error,
                   ByRef LastItem : INTEGER) RETURNS Error
  DECLARE ItemToRemove : Error
  IF LastItem < 0 / = -1
    THEN
      RETURN NullError
    ELSE
      ItemToRemove ← ErrorArray[LastItem]
      LastItem ← LastItem - 1
      RETURN ItemToRemove
    ENDIF
  ENDFUNCTION
```

- (iv) The errors that have been processed are stored in a **global queue**, `ErrorComplete`.

The function `Enqueue` adds an error to `ErrorComplete`:

`Enqueue(ErrorToAdd)`

`Enqueue()` returns `TRUE` if the error is successfully added to the queue and returns `FALSE` if the queue is full.

The procedure `RunError()` should:

- remove an error from the stack using the function `RemoveItem()`
- output message "Stack empty" if there were no errors in the stack
- if an error is returned, add the error to the queue using the function `Enqueue()`
- if the error is added to the queue, output message "Item added to queue"
- if the error is not added to the queue, output message "Item not added to queue".

Copy and complete the pseudocode procedure `RunError()`.

[5]

Ans:

- 1 mark per bullet point to max 5

- Using `RemoveItem(ErrorArray, LastItem)` **and** storing return value ...
- ...checking if return value is `NullError` **and** outputting "stack empty" message if it is null
- ... (if not `NullError`), calling `Enqueue` with return value ...
- ... if return value is `TRUE`, output "added to queue" message ...
- ... if return value is `FALSE` output "not added to queue" message

```
PROCEDURE RunError(BYREF ErrorComplete : ARRAY[0:99] OF Error,
                  BYREF ErrorArray : ARRAY[0:99] OF Error,
                  BYREF LastItem : INTEGER)
  DECLARE DataItem : error
  DataItem ← RemoveItem(ErrorArray, LastItem)
  IF DataItem = NullError THEN
    OUTPUT "Stack empty"
  ELSE
    IF Enqueue(DataItem) = True THEN
      OUTPUT "Item added to queue"
    ELSE
      OUTPUT "Item not added to queue"
    ENDIF
  ENDIF
```

```

ENDIF
ENDPROCEDURE

```

- (c) The stack can be implemented as a linked list with 2D arrays `ErrorArray` with eight (8) nodes. The first column stores Error objects, the second column stores pointer to next object. Pointers should initialize to -1.

Identifiers `ErrorListHead`, `FreeSpaceListHead` are used as the linked list head and free space list head respectively.

- (i) Copy the following diagram and complete the linked list with `ErrorListHead` and `FreeSpaceListHead` and other relevant variables. The first five nodes were inserted with the following errors:

Ans:

LastItem	5 (1)		
ErrorListHead	1 (1)	index	ErrorArray
		1	304-XX 2
		2	208-XY 3
		3	301-AB 4
		4	411-PX 5
		5	709-ZA -1 (1)
FreeSpaceListHead	6 (1)	6	
		7	
		8	-1 (1)

- (ii) Show the changes with another diagram of the linked list after an error removed from the stack. [4]

Ans:

LastItem	4 (1)		
ErrorListHead	1	index	ErrorArray
		1	304-XX 2
		2	208-XY 3
		3	301-AB 4
		4	411-PX -1 (1)
		5	709-ZA -1 (1)
FreeSpaceListHead	6	6	
		7	
		8	5 (1)

- (iii) Explain what the free space list is? [2]

Ans: It is a linked list data structure specifically designed to **keep track** of **unallocated** or **"free"** blocks of array element or memory space. (2)

- 4 Name and explain three different types of errors that could occur in a computer program. Given an example of each type of error with Python programming language. [3]

Ans:		
Errors	Example	
Syntax		(1)
An error that occurs when a command does not follow the expected syntax of the language	<pre> >>> prin("Hello") Traceback (most recent call last): File "<pyshell#0>", line 1, in <module> prin("Hello") NameError: name 'prin' is not defined >>> >>> a = 10 >>> if a == 10 SyntaxError: invalid syntax >>> </pre>	

	>>> print ('DHS') SyntaxError: EOL while scanning string literal	
Runtime or Execution		(1)
An error that only occurs when the program is running and is difficult to foresee before a program is compiled and run	>>> age/0 Traceback (most recent call last): File "<pyshell#6>", line 1, in <module> age/0 ZeroDivisionError: division by zero	
Logical		(1)
An error that causes a program to output an incorrect answer. A logic error occurs when the programmer makes a mistake in their logic for some part of the program.	count = count – 1 should be count = count + 1 call the wrong sub-routine looping too many times	

5 Merge Sort and Insertion Sort are two distinct sorting algorithms with different performance characteristics, particularly evident when analyzed using Big-O notation.

With selected dataset sizes (2^9 , 2^{10} , 2^{11} , 2^{13} , 2^{14} , 2^{15}) comparing these two sorts in terms of worst-case time complexity for **large** datasets. [6]

Ans:
For Merge sort, $O(N \log_2 N)$ in all cases (best, average, and worst). This is because Merge Sort consistently divides the array into halves and then merges them, leading to a logarithmic number of levels of recursion and linear time for merging at each level. (1)

For Insertion sort, Worst Case: $O(N^2)$ when the array is sorted in reverse order, requiring many comparisons and shifts for each element. (1)

Merge Sort's $O(N \log N)$ time complexity makes it significantly more efficient than Insertion Sort's $O(N^2)$ for large datasets. The quadratic growth of Insertion Sort's time complexity means its performance deteriorates rapidly as the input size increases. (2)

Table 1 Comparing $N \log_2 N$ and N^2 (2)		
N	$N \log_2 N$	N^2
512	4,608	262,144
1,024	10,240	1,048,576
2,048	22,458	4,194,304
8,192	106,496	67,108,864
16,384	229,376	268,435,456
32,768	491,520	1,073,741,824

6 A computer uses the ASCII character set.

(a) State the number of characters that can be represented by the ASCII character set. [2]

Ans: number of characters = $2^7 = 128$

(b) Unicode is a different character set.

The Unicode value for the character '1' is denary value 49.

(i) Write the hexadecimal value for the Unicode character '1'. Show your working. [1]

Ans: 16 49 49 divided by 16
Quotient 3 Remainder 1
 $49_{10} = 31_6$

(ii) Write the denary value for the Unicode character '5'. Show your working. [1]

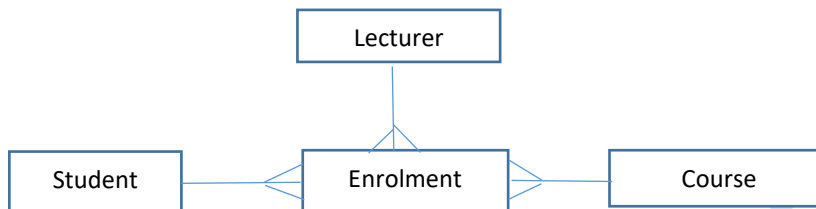
Ans: Denary value of character '1' = 49

Denary value of character '5' = $49 + 4 = 53$ [Or $49 + (5-1)$]

7. A university wishes to store its data using a relational database.

Draw an entity-relationship (ER) diagram to show the tables in third normal form (3NF) and the relationship(s) between them. [3]

Ans:



Answer: 1 mark for each relationship

(a) Write table definitions, indicating the primary key with underline and foreign key with *, for each of the tables you identified in (a). [4]

Use the format: TableName (Attribute1, Attribute2, Attribute3*, etc.)

Ans:

1 mark for each table

Student (StudentID, Name, Email)

Lecturer (LecturerID, Name, Department, Email)

Course (CourseID, CourseName, Department, Credits)

Enrolment (StudentID*, LecturerID*, CourseID*, Semester, Score)

(b) Three lecturers teach the course CS202 in Semester 2025Sem1. After moderation, they decided to add 5 marks to all the students taking the course that semester. Write an SQL query to update the scores. [3]

```
UPDATE Enrolment SET Score = Score + 5 WHERE CourseID = "CS202" AND Semester = "2025Sem1"
```

Ans:

Correct Table to update – 1 mark

Adding 5 marks to Score field – 1 mark

Correct WHERE condition – 1 mark

(c) A student requires at least 100 accumulated credits to graduate. At the end of each semester, the university will generate a report to calculate the total number of credits each student has completed (you can assume that all students pass all of their courses).

Write an SQL query that will output all the students who have completed at least 100 credits. [4]

```
SELECT Student.StudentID, SUM(Credits) FROM Student
INNER JOIN Enrolment ON Student.StudentID = Enrolment.StudentID
INNER JOIN Course ON Enrolment.CourseID = Course.CourseID
GROUP BY Student.StudentID
HAVING SUM(Credits) >= 100;
```

Ans:

Correct fields: Student.StudentID & SUM(Credits) [1 marks]

INNER JOINS with correct tables [1 mark]

GROUP BY [1 mark]

HAVING [1 mark]

(d) State the two aims of normalisation.

[2]

Ans:

1. Reduces redundant data so as to optimise storage space.
2. Prevent data inconsistency due to insert, update or delete anomalies.

(e) The university is considering whether to move its data to a NoSQL database.

(i) State two key differences between the two types of databases.

[2]

SQL	NoSQL
Has fixed, predefined schema	No predefined schema, thus dynamic and can change easily
Data are stored in tables with a fixed data type in each field	Data are stored as collections of documents with no fixed data types
Joins are used to get data across tables, thus easier to use for complex queries	No join operations

Answer: Any two answers

(ii) State and explain which database system the university should choose.

[2]

Possible answer in SQL's favour:

- Since the data stored has a fixed schema and flexibility afforded by NOSQL is unnecessary,
- Complex queries involving join operations are required to retrieve data across multiple tables
- ACID compliance is important
- There is possibly a high number of simultaneous transactions performed during mark entry period

Possible answer in NOSQL's favour:

- NoSQL databases can take advantage of multiple servers, ensuring the application functions even if one server fails
- NoSQL provides flexibility in adding/removing data fields should the requirements of the application change with time
- Horizontally scalable, hence the university might find it easier to add an additional low cost server if the need arises, as compared to replacing existing server with a better one

Answer: Any two reasons for selected database system