සියලු ම හිමිකම් ඇවිරිණි /All Rights Reserved]

Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc Ravindu Bandaranayake - ictfromabc

Information and Communication Technology
තොරතුරු හා සන්නිවේදන තාක්ෂණය

2026 QUIZ NO 442 MARKING

2026/02/25
ictfromabc
Ravindu Bandaranayake

1. Which of the following is located outside the Microprocessor?
   ක්ෂුද්‍ර සකසනයෙන් පිටත පිහිටා ඇත්තේ පහත ඒවායින් කුමක් ද?

   1) ALU. / අංක ගණිතමය හා තාර්කික ඒකකය.
   2) Regiseters. / රෙජිස්තර.
   3) RAM. / සසම්භාවී ප්‍රවේශ මතකය.
   4) Level 1 Cache Memory. / පළමු මට්ටමේ වාරක මතකය.
   5) Control Unit. / පාලන ඒකකය.

   Answer : 3)

   Option / විකල්ප 1)
   The Arithmetic Logic Unit (ALU) is a key component located inside the microprocessor. It performs fundamental arithmetic operations, such as addition and subtraction, as well as logical operations like comparisons. The ALU is essential for executing instructions and processing data, making it a core part of the CPU's functionality.
   අංක ගණිත තාර්කික ඒකකය (ALU) යනු ක්ෂුද්‍ර සකසනය තුළ පිහිටා ඇති ප්‍රධාන අංගයකි. එය එකතු කිරීම සහ අඩු කිරීම වැනි මූලික ගණිතමය මෙහෙයුම් මෙන්ම සංසන්දනය වැනි තාර්කික මෙහෙයුම් සිදු කරයි. උපදෙස් ක්‍රියාත්මක කිරීමට සහ දත්ත සැකසීමට ALU අත්‍යවශ්‍ය වන අතර එය CPU හි ක්‍රියාකාරීත්වයේ මූලික කොටසක් බවට පත් කරයි.

   Option / විකල්ප 2)
   Registers are small, high-speed storage locations that are also found inside the microprocessor. They temporarily hold data and instructions that the CPU is currently processing. Because registers are faster than other types of memory, they help optimize performance by allowing the CPU to access data quickly without delays from slower memory sources.
   රෙජිස්ටර් යනු ක්ෂුද්‍ර සකසනය තුළ ද ඇති කුඩා, අධිවේගී ගබඩා ස්ථාන වේ. CPU දැනට සකසන දත්ත සහ උපදෙස් ඔවුන් තාවකාලිකව රඳවා තබා ගනී. රෙජිස්ටර් අනෙකුත් මතක වර්ග වලට වඩා වේගවත් බැවින්, මන්දගාමී මතක ප්‍රභවයන්ගෙන් ප්‍රමාදයකින් තොරව ඉක්මනින් දත්ත වෙත ප්‍රවේශ වීමට CPU හට ඉඩ දීමෙන් කාර්ය සාධනය ප්‍රශස්ත කිරීමට ඒවා උපකාරී වේ.

   Option / විකල්ප 3)
   Random Access Memory (RAM) is a type of volatile memory located outside the microprocessor. It serves as the main working memory for the CPU, storing data and instructions that are actively in use. RAM is essential for running programs and processes, and its speed and capacity directly impact overall system performance.
   සසම්භාවී ප්‍රවේශ මතකය (RAM) යනු ක්ෂුද්‍ර සකසනයෙන් පිටත පිහිටි නශ්‍ය මතක වර්ගයකි. එය CPU සඳහා ප්‍රධාන ක්‍රියාකාරී මතකය ලෙස ක්‍රියා කරයි, සක්‍රියව භාවිතයේ පවතින දත්ත සහ උපදෙස් ගබඩා කරයි. වැඩසටහන් සහ ක්‍රියාවලි ක්‍රියාත්මක කිරීම සඳහා RAM අත්‍යවශ්‍ය වන අතර එහි වේගය සහ ධාරිතාව සමස්ත පද්ධතියේ ක්‍රියාකාරිත්වයට සෘජුවම බලපායි.

   Option / විකල්ප 4)
   Level 1 Cache Memory is a small, ultra-fast memory cache located inside the microprocessor. It stores frequently accessed data and instructions to reduce latency and speed up processing. By providing the CPU with quick access to critical information, Level 1 cache enhances overall system efficiency and performance.
   Level 1 Cache Memory යනු ක්ෂුද්‍ර සකසනය තුළ පිහිටා ඇති කුඩා, අතිශය වේගවත් මතක වාරකයකි. එය ප්‍රමාදය අඩු කිරීමට සහ සැකසීම වේගවත් කිරීමට නිතර ප්‍රවේශ වන දත්ත සහ උපදෙස් ගබඩා කරයි. CPU වෙත තීරණාත්මක තොරතුරු වෙත ඉක්මන් ප්‍රවේශයක් ලබා දීමෙන්, Level 1 cache මඟින් සමස්ත පද්ධතියේ කාර්යක්ෂමතාව සහ කාර්ය සාධනය වැඩි දියුණු කරයි.

Option / විකල්ප 5)
The Control Unit is another vital component located within the microprocessor. It orchestrates the operations of the CPU by directing how instructions are executed and coordinating the activities of the ALU, registers, and memory. The Control Unit interprets instructions fetched from memory and generates control signals, ensuring that all parts of the microprocessor work together seamlessly.
Control Unit යනු ක්ෂුද්‍ර සකසනය තුළ පිහිටා ඇති තවත් වැදගත් අංගයකි. උපදෙස් ක්‍රියාත්මක වන ආකාරය සහ ALU රෙජිස්ටර් සහ මතකයේ ක්‍රියාකාරකම් සම්බන්ධීකරණය කිරීම මගින් එය CPU හි මෙහෙයුම් සංවිධානය කරයි. පාලන ඒකකය මතකයෙන් ලබාගත් උපදෙස් අර්ථකථනය කර පාලන සංඥා ජනනය කරයි, ක්ෂුද්‍ර සකසනයේ සියලුම කොටස් බාධාවකින් තොරව එකට ක්‍රියා කරන බව සහතික කරයි.

Therefore, the correct answer is 3) RAM.
එබැවින්, නිවැරදි පිළිතුර 3) RAM වේ.

2. What is the 8 bit 2's complement representation of –57 ?
-57 හි 8 bit 2 හි අනුපූරක නිරූපණය කුමක්ද?

1) <u>11000111_2</u> 2) 11000110_2 3) 00111001_2 4) 01000111_2 5) 00111000_2

Answer : 1)

First Let's find the 8 bit binary representation of 57 and -57 මුලින්ම අපි 57 සහ -57 හි බිටු 8 ද්විමය නිරූපණය සොයා ගනිමු							
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	1	1	1	0	0	1
57 = 00111001_2							
-57 = 11000110_2							

Now Let's Find the 2's complement of -57 දැන් අපි -57 හි 2 හි අනුපූරකය සොයා ගනිමු							
1	1	0	0	0	1	1	0
						+	1
1	1	0	0	0	1	1	1
2's complement of -57 = 11000111_2							

Therefore the correct answer is 1).
එබැවින් නිවැරදි පිළිතුර 1) වේ.

3. Which of the following answer is equivalent to this Boolean statement?
පහත සඳහන් කුමන පිළිතුර මෙම බූලියානු ප්‍රකාශයට සමානද?

$(CB + DB)B + \overline{CC}$

1) 1 2) 0 3) B 4) <u>$B+\overline{C}$</u> 5) D.A

Answer : 4)

This boolean expression can be simplfied as follows.
මෙම බූලියානු ප්‍රකාශනය පහත ආකාරයට සුළු කළ හැකිය.

$(CB + DB)B + \overline{CC}$	
$(CB + DB)B + \overline{C} + \overline{C}$	Demorgan Therom / ද මුවාර න්‍යාය
$(CB + DB)B + \overline{C}$	Idempotent Law / තදේවභාවි න්‍යාය
$BCB + BDB + \overline{C}$	Distribution / විබේදන න්‍යාය
$BC + BD + \overline{C}$	Idempotent Law / තදේවභාවි න්‍යාය
$B + BD + \overline{C}$	Redundancy Law / සමතිරික්ත න්‍යාය
$B + \overline{C}$	Redundancy Law / සමතිරික්ත න්‍යාය

Therefore the correct answer is **4). /** එබැවින් නිවැරදි පිළිතුර **4)** වේ.

4. Which of the following is not true about time sharing operating systems?
කාල විභජන මෙහෙයුම් පද්ධති සම්බන්ධයෙන් පහත සඳහන් ඒවායින් සත්‍ය නොවන්නේ කුමක්ද?

1) They enhance resource utilization by sharing resources.
සම්පත් බෙදාගැනීමෙන් සම්පත් භාවිතය වැඩි දියුණු කරයි.

2) They allocate CPU time slices to multiple users.
ඒවා බහු පරිශීලකයින් සඳහා CPU කාල රාමු වෙන් කරයි.

3) They rely on context switching to manage sessions.
සැසි කළමනාකරණය කිරීමට සන්දර්භ ස්විචනය මත රඳා පවතී.

4) They provide quick response times for interactive tasks.
අන්තර්ක්‍රියාකාරී කාර්යයන් සඳහා ඉක්මන් ප්‍රතිචාර දැක්වීමේ කාල සපයයි.

5) <u>A single user can monopolize the CPU for long periods.</u>
තනි පරිශීලකයෙකුට දිගු කාලයක් සඳහා CPU ඒකාධිකාරය කළ හැක.

Answer : 5)

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

Option / විකල්ප 1)
True. Time-sharing operating systems are designed to improve resource utilization by allowing multiple users to share resources effectively.
සත්‍ය වේ. කාල විභාජන මෙහෙයුම් පද්ධති සැලසුම් කර ඇත්තේ බහු පරිශීලකයින්ට සම්පත් ඵලදායී ලෙස බෙදා ගැනීමට ඉඩ දීමෙන් සම්පත් භාවිතය වැඩිදියුණු කිරීම සඳහා ය.

Option / විකල්ප 2)
True. Time-sharing systems allocate CPU time in small slices to different users, ensuring fair access.
සත්‍ය වේ. කාල විභාජන පද්ධති සාධාරණ ප්‍රවේශය සහතික කරමින් විවිධ පරිශීලකයින්ට කුඩා කොටස් (slices) වලට CPU කාලය වෙන් කරයි.

Option / විකල්ප 3)
True. Context switching is essential in time-sharing systems, enabling the system to switch between users' tasks efficiently.
සත්‍ය වේ. කාල විභාජන පද්ධතිවල සන්දර්භය මාරු කිරීම අත්‍යවශ්‍ය වන අතර, පරිශීලකයින්ගේ කාර්යයන් අතර කාර්යක්ෂමව මාරු වීමට පද්ධතියට හැකි වේ.

Option / විකල්ප 4)
True. Time-sharing systems aim to provide quick response times, especially for interactive tasks, by rapidly switching between users.
සත්‍ය වේ. පරිශීලකයන් අතර වේගයෙන් මාරු වීමෙන්, විශේෂයෙන් අන්තර්ක්‍රියාකාරී කාර්යයන් සඳහා ඉක්මන් ප්‍රතිචාර දැක්වීමේ වේලාවන් සැපයීම කාල විභාජන පද්ධතිවල අරමුණයි.

Option / විකල්ප 5)
False. Time-sharing systems are designed to prevent any single user from monopolizing the CPU for long periods by dividing CPU time into small slices across users.
අසත්‍ය වේ. කාල විභාජන පද්ධති සැලසුම් කර ඇත්තේ ඕනෑම තනි පරිශීලකයෙකු විසින් CPU කාලය භාවිතා කරන්නන් හරහා කුඩා කොටස් (slices) වලට බෙදීම මගින් දිගු කාලයක් සඳහා CPU ඒකාධිකාරී කිරීම වැලැක්වීමටය.

Therefore, the correct answer is option 5).
එබැවින්, නිවැරදි පිළිතුර විකල්ප 5) වේ.

5. Refer the given tables and find out which of the following SQL queries will give an output that displays the department names along with the count of employees whose salary is greater than 55,000.
දී ඇති වගු පරිශීලනය කර, පහත සඳහන් කුමන SQL විමසුම් මඟින් දෙපාර්තමේන්තු නාම සහ වැටුප 55,000 ට වඩා වැඩි සේවකයින් සංඛ්‍යාව පෙන්වන ප්‍රතිදානයක් ලබා දෙන්නේ දැයි සොයන්න.

Employees Table

EmpID	Name	DeptID	Salary
101	Alice	1	60000
102	Bob	2	56000
103	Charlie	1	70000
104	David	3	50000

Departments Table

DeptID	DeptName
1	HR
2	IT
3	Finance

Expected Output: / අපේක්ෂිත ප්‍රතිදානය:

DeptName	NumEmployees
HR	2
IT	1

1)
```
SELECT d.DeptName, COUNT(e.EmpID) AS NumEmployees
FROM Departments d
JOIN Employees e ON d.DeptID = e.DeptID
WHERE e.Salary > 55000
GROUP BY d.DeptName;
```

2)
```
SELECT d.DeptName, COUNT(e.EmpID) AS NumEmployees
FROM Departments d
LEFT JOIN Employees e ON d.DeptID = e.DeptID
WHERE e.Salary > 55000
GROUP BY d.DeptName;
```

3)
```
SELECT d.DeptName, COUNT(e.EmpID) AS NumEmployees
FROM Employees e
RIGHT JOIN Departments d ON e.DeptID = d.DeptID
WHERE e.Salary > 55000
GROUP BY d.DeptName;
```

4)
```
SELECT d.DeptName, COUNT(e.EmpID) AS NumEmployees
FROM Departments d
LEFT JOIN Employees e ON d.DeptID = e.DeptID
GROUP BY d.DeptName
HAVING SUM(e.Salary) > 55000;
```

5)
```
SELECT d.DeptName, COUNT(e.EmpID) AS NumEmployees
FROM Departments d
JOIN Employees e ON d.DeptID = e.DeptID
GROUP BY d.DeptName
HAVING COUNT(CASE WHEN e.Salary >= 55000 THEN 1 END) > 0;
```

Answer : 1)

We need to find the department names along with the count of employees who have a salary greater than 55,000.
අපි දෙපාර්තමේන්තු නම් සහ 55,000 ට වැඩි වැටුපක් ලබන සේවකයින්ගේ ගණන සොයා ගත යුතුයි.

1. Filtering employees with salary > 55,000 / 55,000 ට වැඩි වැටුපක් ඇති සේවකයින් තෝරාගැනීම

- From the Employees table, we select only those rows where Salary > 55000.
  Employee table එකෙන් අපි වැටුප > 55000 ඇති පේළි පමණක් තෝරා ගනිමු.

2. Joining with Departments table / Department table සමඟ සම්බන්ධ කිරීම

- Then we need to join the filtered employees with the Departments table on DeptID.
  ඉන්පසු අපි DeptID හි දෙපාර්තමේන්තු වගුව සමඟ තෝරාගත් සේවකයින් සමඟ සම්බන්ධ විය යුතුය.

Option A - Correct / නිවැරදියි

- JOIN ensures only matching records are retrieved.
  JOIN ගැලපෙන වාර්තා පමණක් ලබා ගැනීම සහතික කරයි.

- WHERE e.Salary > 55000 filters employees before counting.
  WHERE e.Salaray > 55000 සේවකයින් ගණන් කිරීමට පෙර තෝරාගනී.

- GROUP BY d.DeptName groups by department.
  GROUP BY d.DeptName දෙපාර්තමේන්තුව අනුව කණ්ඩායම් කරයි.

Option B - Incorrect / වැරදිය

- While it gives the correct result in this case, using LEFT JOIN is unnecessary.
  මෙම අවස්ථාවේදී එය නිවැරදි ප්‍රතිඵලය ලබා දුන්නද, LEFT JOIN භාවිතා කිරීම අනවශ්‍යය.

Option C - Incorrect / වැරදිය

- RIGHT JOIN is not needed, as we are only interested in departments with employees earning more than 55,000.
  RIGHT JOIN අවශ්‍ය නොවේ, මන්ද අපි උනන්දු වන්නේ රු. (55,000) ට වඩා වැඩි වැටුප් ලබන සේවකයින් සිටින දෙපාර්තමේන්තු සඳහා පමණි.

Option D - Incorrect / වැරදිය

- HAVING SUM(e.Salary) > 55000 sums all salaries instead of filtering individual employees.
  HAVING SUM(e.Salary) > 55000 තනි සේවකයින් තේරීම වෙනුවට සියලු වැටුප් එකතු කරයි.

Option E - Incorrect / වැරදිය

- This condition (e.Salary >= 55000) includes salaries equal to 55,000.
  මෙම කොන්දේසියට (e.Salary >= 55000) ට සමාන වැටුප් ද ඇතුළත් වේ.

- If there were an employee earning exactly 55,000, they would incorrectly be counted.
  හරියටම රු. 55,000 ක් උපයන සේවකයෙකු සිටියේ නම්, ඔවුන් වැරදි ලෙස ගණන් ගනු ලැබේ.

Therefore, the correct answer is 1).
එබැවින්, නිවැරදි පිළිතුර 1) වේ.

6. Consider the below python code.
   පහත පයිතන් කේතය සලකා බලන්න.

```
scores = [32, 45, 51, 60, 74, 85, 91]
for score in scores:
    if score % 5 == 0:
        continue
    print(score)
```

What will be the output? / ප්‍රතිදානය කුමක් වේද?

1)	2)	3)	4)	5)
32	32	32	45	Error
51	45	51	60	
74	51	60	85	
91	60	74		
	74	85		
	85			
	91			

Answer : 1)

The given Python code iterates through the scores list and checks each value. If a score is a multiple of 5 (such as 45, 60, and 85), the continue statement skips printing that value. The remaining scores (32, 51, 74, and 91) are printed because they are not divisible by 5.
දී ඇති පයිතන් කේතය scores ලැයිස්තුව හරහා පුනරාවර්තනය වී එක් එක් අගය පරීක්ෂා කරයි. ලකුණු 5 න් ගුණාකාරයක් නම් (45, 60 සහ 85 වැනි), continue ප්‍රකාශය එම අගය මුද්‍රණය කිරීම මඟ හරියි. ඉතිරි ලකුණු (32, 51, 74, සහ 91) මුද්‍රණය කර ඇත්තේ ඒවා 5 න් බෙදිය නොහැකි බැවිනි.

Step-by-step Explanation / පියවරෙන් පියවර විස්තර කිරීම

The list scores contains the values: `[32, 45, 51, 60, 74, 85, 91]`.
scores ලැයිස්තුව අගයන්ගෙන් අඩංගු වේ : `[32, 45, 51, 60, 74, 85, 91]`.

A for loop iterates through each score in the list.
for ලූපය ලැයිස්තුවේ එක් එක් ලකුණු හරහා පුනරාවර්තනය වේ.

The if statement checks whether the score is a multiple of 5 (`score % 5 == 0`).
if ප්‍රකාශය ලකුණු 5 හි ගුණාකාරයක් දැයි පරීක්ෂා කරයි (`score % 5 == 0`)

1. If True, the continue statement is executed, skipping that iteration.
   සත්‍ය නම්, continue ප්‍රකාශය ක්‍රියාත්මක කරනු ලැබේ, එම පුනරාවර්තනය මඟ හරිනු ලැබේ.
2. If False, the score is printed.
   අසත්‍ය නම්, ලකුණු මුද්‍රණය කෙරේ.

The multiples of 5 in the list are 45, 60, and 85, so these values are skipped.
ලැයිස්තුවේ 5 හි ගුණාකාර 45, 60 සහ 85 වන බැවින් මෙම අගයන් මඟ හරිනු ලැබේ.
The remaining scores (32, 51, 74, 91) are printed.
ඉතිරි ලකුණු (32, 51, 74, 91) මුද්‍රණය කර ඇත.

Therefore, the correct output is `32, 51, 74, 91`, making option 1 the correct answer.
එබැවින්, නිවැරදි ප්‍රතිදානය `32, 51, 74, 91` වන අතර, විකල්පය 1 නිවැරදි පිළිතුර බවට පත් කරයි.

7. Which of the following is the correct way to reverse a list and get the output in Python?
   පයිතන් හි ලැයිස්තුවක් ආපසු හරවා ප්‍රතිදානය ලබාගැනීමට පහත සඳහන් ඒවායින් නිවැරදි ක්‍රමය කුමක්ද?

   1) `list.reverse()`
   2) `reverse(list)`
   3) `list[::-1]`
   4) `reversed(list)`
   5) `list.reverse() and list`

Answer : 3)

Explanation / පැහැදිලි කිරීම

To reverse a list in Python, one of the most efficient and commonly used methods is slicing with `list[::-1]`. This slicing notation works by specifying the start, stop, and step parameters, where `[::-1]` tells Python to traverse the list from the end to the beginning, effectively creating a reversed copy of the list. It does not modify the original list but instead returns a new one with the elements reversed. This makes it ideal when you want to keep the original list intact and require a reversed version for further processing.
Python හි ලැයිස්තුවක් ආපසු හැරවීම සඳහා, වඩාත් කාර්යක්ෂම හා බහුලව භාවිතා වන ක්‍රමවලින් එකක් වන්නේ `list[::-1]` සමඟ slicing වේ. මෙම slicing අංකනය ආරම්භක, නැවතුම් සහ පියවර පරාමිතීන් නියම කිරීමෙන් ක්‍රියා කරයි, එහිදී `[::-1]` Python හට ලැයිස්තුව අවසානයේ සිට ආරම්භය දක්වා ගමන් කරන ලෙස පවසන අතර, ලැයිස්තුවේ ප්‍රතිලෝම පිටපතක් ඵලදායී ලෙස නිර්මාණය කරයි. එය මුල් ලැයිස්තුව වෙනස් නොකරයි, නමුත් ඒ වෙනුවට මූලද්‍රව්‍ය ප්‍රතිලෝම කර නව එකක් ආපසු ලබා දෙයි. ඔබට මුල් ලැයිස්තුව නොවෙනස්ව තබා ගැනීමට අවශ්‍ය වූ විට සහ තවදුරටත් සැකසීම සඳහා ප්‍රතිලෝම අනුවාදයක් අවශ්‍ය වූ විට මෙය වඩාත් සුදුසු වේ.

Other options for reversing a list include `list.reverse()` and `reversed(list)`. The `list.reverse()` method modifies the original list in place, which is not always desirable when the original order must be preserved. On the other hand, `reversed(list)` returns an iterator that generates elements in reverse order, which requires explicit conversion (e.g., `list(reversed(list))`) to get a new list. While both methods are valid, using `list[::-1]` is more concise and elegant, especially when a new reversed list is needed without affecting the original.
ලැයිස්තුවක් ආපසු හැරවීම සඳහා වෙනත් විකල්ප අතර `list.reverse()` සහ `reversed(list)` ඇතුළත් වේ. `list.reverse()` ක්‍රමය මුල් ලැයිස්තුව ස්ථානයේ වෙනස් කරයි, මුල් අනුපිළිවෙල සංරක්ෂණය කළ යුතු විට එය සැමවිටම යෝග්‍ය නොවේ. අනෙක් අතට, `reversed(list)` ප්‍රතිලෝම අනුපිළිවෙලින් මූලද්‍රව්‍ය ජනනය කරන පුනරාවර්තකයක් ආපසු ලබා දෙයි, ඒ සඳහා පැහැදිලි පරිවර්තනයක් අවශ්‍ය වේ (උදා: `list(reversed(list))`). ක්‍රම දෙකම වලංගු වුවද, `list[::-1]` භාවිතා කිරීම වඩාත් සංක්ෂිප්ත සහ අලංකාර වේ, විශේෂයෙන් මුල් පිටපතට බලපෑමක් නොකර නව ප්‍රතිලෝම ලැයිස්තුවක් අවශ්‍ය වූ විට.

Therefore, the correct answer is, 3). / එබැවින්, නිවැරදි පිළිතුර, 3) වේ.

8. What is the correct way to handle exceptions while interacting with a database in Python?
පයිතන් හි දත්ත සමුදායක් සමඟ අන්තර් ක්‍රියා කරන අතරතුර ව්‍යතිරේක හැසිරවිය හැකි නිවැරදි ක්‍රමය කුමක්ද?

1) <u>Using try-except blocks around database connection and operations.</u>
දත්ත සමුදා සම්බන්ධතාවය සහ මෙහෙයුම් වටා ඇති try-except බ්ලොක් භාවිතා කිරීම.
2) Using only try blocks / try බ්ලොක් පමණක් භාවිතා කිරීම.
3) Using only except blocks / except බ්ලොක් පමණක් භාවිතා කිරීම.
4) Using a separate error-handling module / වෙනම දෝෂ හැසිරවීමේ මොඩියුලයක් භාවිතා කිරීම.
5) Using conditional statements / කොන්දේසි සහිත ප්‍රකාශ භාවිතා කිරීම.

Answer : 1)

Explanation / පැහැදිලි කිරීම

Option / විකල්ප 1)
True. The best practice is to use try-except blocks to catch and handle database-related exceptions. This prevents the program from crashing if an error occurs (e.g., connection failure, SQL syntax error).
නිවැරදියි. හොඳම භාවිතය වන්නේ දත්ත සමුදාය ආශ්‍රිත exceptions අල්ලා ගැනීමට සහ හැසිරවීමට try-except බ්ලොක් භාවිතා කිරීමයි. මෙය දෝෂයක් සිදු වුවහොත් වැඩසටහන බිඳ වැටීම වළක්වයි (උදා: සම්බන්ධතා අසාර්ථකත්වය, SQL සින්ටැක්ස් දෝෂය).

Explanation of Incorrect Options / වැරදි විකල්ප පැහැදිලි කිරීම

Option / විකල්ප 2)
False. A try block must have an except or finally block to handle errors properly. If an error occurs inside a try block without an except, the program will crash.
වැරදියි. දෝෂ නිසියාකාරව හැසිරවීමට try බ්ලොක් හැර හෝ අවසාන අවහිරයක් තිබිය යුතුය. හැරවීමකින් තොරව try බ්ලොක් එකක් තුළ දෝෂයක් ඇති වුවහොත්, වැඩසටහන බිඳ වැටේ.

Option / විකල්ප 3)
False. except blocks cannot be used alone. They must follow a try block to catch exceptions.
වැරදියි. except බ්ලොක් තනිව භාවිතා කළ නොහැක. ව්‍යතිරේක අල්ලා ගැනීමට ඔවුන් උත්සාහක වාරයක් අනුගමනය කළ යුතුය.

Option / විකල්ප 4)
False. Python does not have a built-in separate error-handling module for database exceptions. Error handling should be done using try-except within the script.
වැරදියි. දත්ත සමුදා ව්‍යතිරේක සඳහා පයිතන් සතුව වෙනම දෝෂ හැසිරවීමේ මොඩියුලයක් නොමැත. දෝෂ හැසිරවීම ස්ක්‍රිප්ටය තුළ හැර උත්සාහය භාවිතයෙන් සිදු කළ යුතුය.

Option / විකල්ප 5)
False. Conditional statements (if-else) do not handle exceptions that occur at runtime. They are used for logic control, not for catching database errors.
වැරදියි. කොන්දේසි සහිත ප්‍රකාශ (if-else) ධාවන වේලාවේදී සිදුවන ව්‍යතිරේක හසුරුවන්නේ නැත. ඒවා භාවිතා කරන්නේ තාර්කික පාලනය සඳහා මිස දත්ත සමුදා දෝෂ අල්ලා ගැනීම සඳහා නොවේ.

Therefore the Correct answer is 1) / එබැවින් නිවැරදි පිළිතුර 1)

9. Which of the following statements about HTML is correct?
HTML පිළිබඳ පහත සඳහන් ප්‍රකාශවලින් නිවැරදි කුමක්ද?

1) HTML is a programming language used to create dynamic web applications.
HTML යනු ගතික වෙබ් යෙදුම් නිර්මාණය කිරීමට භාවිතා කරන ක්‍රමලේඛන භාෂාවකි.
2) The <title> tag is used to define the main heading of a webpage.
වෙබ් පිටුවක ප්‍රධාන ශීර්ෂය අර්ථ දැක්වීමට <title> උසුලනය භාවිතා කරයි.
3) The <img> tag requires both a src and alt attribute to function correctly.
<img> උසුලනය නිවැරදිව ක්‍රියා කිරීමට src සහ alt ගුණාංග දෙකම අවශ්‍ය වේ.
4) The <a> tag is used to create hyperlinks in HTML.
HTML හි අධි සබැඳි නිර්මාණය කිරීමට <a> උසුලනය භාවිතා කරයි.
5) HTML is case-sensitive, meaning <Div> and <div> are treated differently.
HTML අක්ෂර සංවේදී වේ, එනම් <Div> සහ <div> වෙනස් ලෙස සලකනු ලැබේ.

Answer: 4)

Explanation of each option / එක් එක් විකල්පය පැහැදිලි කිරීම:

Option / විකල්ප 1)
HTML is not a programming language; it is a markup language used for structuring content on webpages. JavaScript is used for dynamic web applications.
HTML ක්‍රමලේඛන භාෂාවක් නොවේ; එය වෙබ් පිටු වල අන්තර්ගතය ව්‍යුහගත කිරීම සඳහා භාවිතා කරන සලකුණු භාෂාවකි. ගතික වෙබ් යෙදුම් සඳහා JavaScript භාවිතා කරයි.

Option / විකල්ප 2)
The <title> tag defines the title of the webpage, which appears in the browser tab. The <h1> tag is used for the main heading.
<title> උසුලනය අතිරික්සු පටිත්තෙහි දිස්වන වෙබ් පිටුවේ මාතෘකාව අර්ථ දක්වයි. <h1> උසුලනය ප්‍රධාන ශීර්ෂය සඳහා භාවිතා කරයි.

Option / විකල්ප 3)
The <img> tag requires only the src (source) attribute to function, but the alt (alternative text) attribute is recommended for accessibility.
<img> උසුලනය නිවැරදිව ක්‍රියා කිරීමට src (මූලාශ්‍ර) ගුණාංගය පමණක් අවශ්‍ය වේ, නමුත් ප්‍රවේශ්‍යතාව සඳහා alt (විකල්ප පෙළ) ගුණාංගය නිර්දේශ කෙරේ.

Option / විකල්ප 4)
The <a> tag is allowing users to navigate to other webpages or sections within a page.
<a> උසුලනය පරිශීලකයින්ට පිටුවක් තුළ ඇති අනෙකුත් වෙබ් පිටු හෝ කොටස් වෙත සැරිසැරීමට ඉඩ දෙයි.

Option / විකල්ප 5)
HTML is not case-sensitive, so <Div>, <div>, and <DIV> are treated the same.
HTML අකුරු අකුරු වලට සංවේදී නොවන බැවින් <Div>, <div> සහ <DIV> එකම ලෙස සලකනු ලැබේ.

Therefore, the correct answer is 4).
එබැවින්, නිවැරදි පිළිතුර 4) වේ.

10. The following CSS code is example for which type of CSS?
පහත CSS කේතය කුමන ආකාරයේ CSS සඳහා උදාහරණ වේ ද?

```
<p style ="border: 2px solid red ;">
```

1) External CSS
බාහිර CSS

2) Internal CSS
අභ්‍යන්තර CSS

3) Inline CSS
පේළිගත CSS

4) Hyperlink CSS
අධිසම්බන්ධක CSS

5) None of the above
ඉහත කිසිවක් නොවේ

Answer : 3)

Explanation / පැහැදිලි කිරීම

Option / විකල්ප 1)
Incorrect. External CSS involves linking to an external .css file, not embedding styles within HTML tags. වැරදියි. බාහිර CSS යනු බාහිර .css ගොනුවකට සම්බන්ධ කිරීම, HTML උසුලන තුළ ශෛලීන් කාවැද්දීම නොවේ.

Option / විකල්ප 2)
Incorrect. Internal CSS involves defining styles within the <style> tag in the <head> section of the HTML document, not within individual HTML tags.
වැරදියි. අභ්‍යන්තර CSS යනු තනි තනි HTML උසුලන තුළ නොව HTML ලේඛනයේ <head>කොටසේ <style> උසුලනය තුළ මෝස්තර නිර්වචනය කිරීමයි.

Option / විකල්ප 3)
Correct. This is an example of Inline CSS, where the CSS style is applied directly within the HTML tag using the style attribute.
නිවැරදියි. මෙය Inline CSS සඳහා උදාහරණයකි, එහිදී CSS විලාසය HTML උසුලනය තුළ විලාස ගුණාංගය භාවිතයෙන් සෘජුවම යොදනු ලැබේ.

Option / විකල්ප 4)
Incorrect. Hyperlink CSS is not a standard term for a CSS type.
වැරදියි. Hyperlink CSS යනු CSS වර්ගයක් සඳහා සම්මත පදයක් නොවේ.

Therefore, the correct answer is 3)
එබැවින් නිවැරදි පිළිතුර 3)