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 409 MARKING

2026/01/23
ictfromabc
Ravindu Bandaranayake

1. What is the incorrect statement regarding the manual and computerized data input methods?
   පරිගණකගත දත්ත ආදාන ක්‍රම සම්බන්ධයෙන් ඇති වැරදි ප්‍රකාශය කුමක්ද?

   1) accuracy is high / නිරවද්‍යතාවය ඉහළයි.
   2) faster data input / වේගවත් දත්ත ආදානය.
   3) cost effectiveness / පිරිවැය ඵලදායිතාවය.
   4) minimal requirements of data verification / දත්ත සත්‍යාපනයේ අවම අවශ්‍යතා.
   5) compulsory to verify all the data inserted using computerized method.
      පරිගණකගත ක්‍රමය භාවිතයෙන් ඇතුළු කරන ලද සියලුම දත්ත තහවුරු කළ යුතුය.

   Answer : 5)

   Explanation / පැහැදිලි කිරීම

   The statement "compulsory to verify all the data inserted using a computerized method" is incorrect regarding manual and computerized data input methods. In computerized methods, data is often subjected to automated validation and error-checking processes during input, which minimizes the need for manual verification of every entry. This automation enhances efficiency and reduces human intervention, making it unnecessary to verify every piece of data manually.
   අත්හුරු සහ පරිගණකගත දත්ත ආදාන ක්‍රම සම්බන්ධයෙන් "පරිගණකගත ක්‍රමයක් භාවිතයෙන් ඇතුලත් කරන ලද සියලුම දත්ත සත්‍යාපනය කිරීම අනිවාර්යය" යන ප්‍රකාශය වැරදිය. පරිගණකගත ක්‍රම වලදී, දත්ත බොහෝ විට ස්වයංක්‍රිය වලංගුකරණයට සහ ආදානය අතරතුර දෝෂ පරීක්ෂා කිරීමේ ක්‍රියාවලින්ට යටත් වන අතර එමඟින් සෑම ප්‍රවේශයක්ම අතින් සත්‍යාපනය කිරීමේ අවශ්‍යතාවය අවම කරයි. මෙම ස්වයංක්‍රියකරණය කාර්යක්ෂමතාව වැඩි කරන අතර මානව මැදිහත්වීම අඩු කරයි, සෑම දත්ත කැබැල්ලක්ම අතින් සත්‍යාපනය කිරීම අනවශ්‍ය කරයි.

   On the other hand, manual data input methods typically require more rigorous verification since errors are more likely to occur due to human mistakes. While computerized methods aim to enhance accuracy and reduce the need for verification, stating that verification is compulsory for all data entered contradicts the efficiency and accuracy benefits that computerized systems are designed to provide. This makes the fifth statement incorrect in the context of these methods.
   අනෙක් අතට, අත්හුරු දත්ත ආදාන ක්‍රමවලට සාමාන්‍යයෙන් වඩාත් දැඩි සත්‍යාපනයක් අවශ්‍ය වේ, මන්ද මිනිස් වැරදි නිසා දෝෂ ඇති වීමට ඉඩ ඇත. පරිගණක ගත ක්‍රමවල නිරවද්‍යතාවය වැඩි දියුණු කිරීම සහ සත්‍යාපනය සඳහා අවශ්‍යතාවය අඩු කිරීම අරමුණු කර ඇති අතර, ඇතුළත් කරන ලද සියලුම දත්ත සඳහා සත්‍යාපනය අනිවාර්ය බව ප්‍රකාශ කිරීම පරිගණක ගත පද්ධති ලබා දීමට සැලසුම් කර ඇති කාර්යක්ෂමතාව සහ නිරවද්‍යතා ප්‍රතිලාභවලට පටහැනි වේ. මෙම ක්‍රමවල සන්දර්භය තුළ පස්වන ප්‍රකාශය වැරදියි.

   So, the correct answer number is 5).
   එමනිසා, නිවැරදි පිළිතුරු අංකය 5) වේ.

2. What is the decimal equivalent of the hexadecimal number $BC12_{16}$?
$BC12_{16}$ ෂඩ් දශම සංඛ්‍යාවට සමාන දශමය අගය කුමක්ද?

1) 47146 2) 46126 3) 48146 4) 49116 5) 45036

Answer : 3)

Let's convert the hexadecimal number $BC12_{16}$ into decimal.
$BC12_{16}$ ෂඩ්දශමය අංකය දශමයට පරිවර්තනය කරමු.

$BC12_{16}$			
B	C	1	2
B = 11	C = 12	1	2
$16^3 = 4096$	$16^2 = 256$	$16^1 = 16$	$16^0 = 1$
4096 × 11	256 × 12	16 × 1	1 × 2
45056	3072	16	2
45056 + 3072 + 16 + 2			
48146_{10}			

Therefore, the answer is 3).
එබැවින්, පිළිතුර වන්නේ 3) වේ.

3. (P NOR Q) NOR (R NAND P) OR (P AND Q)

If P = True , Q = True , R = False then the corresponding answer is,
P = True, Q = True, R = False නම් අදාළ පිළිතුර වන්නේ,

1) 1 2) P.Q.R 3) True 4) False 5) P

Answer : 3)

The following expression can be obtained by substituting the values True and False for P, Q, and R in the above boolean expression.
ඉහත බූලිය ප්‍රකාශනයේ P, Q හා R සඳහා True හා False යන අගයන් ආදේශ කර පහත ප්‍රකාශනය ලබා ගත හැක.
(True NOR True) NOR (False NAND True) OR (True AND True)

In a NOR gate, when both inputs are True, the output is False.
NOR ද්වාරයක් තුළදී ආදාන ද්විත්වයම සත්‍ය වන අවස්ථාවේදී ප්‍රතිදානය අසත්‍ය වේ.
(False) NOR (False NAND True) OR (True AND True)

In a NAND gate, when both inputs are not equal, the output is True.
NAND ද්වාරයක් තුළදී ආදාන ද්විත්වයම අසමාන වන අවස්ථාවේදී ප්‍රතිදානය සත්‍ය වේ.
(False) NOR (True) OR (True AND True)

In an AND gate, when both inputs are True, the output is True.
AND ද්වාරයක් තුළදී ආදාන ද්විත්වයම සත්‍ය වන අවස්ථාවේදී ප්‍රතිදානය සත්‍ය වේ.
(False) NOR (True) OR (True)

In a NOR gate, when both inputs are not equal, the output is False.
NOR ද්වාරයක් තුළදී ආදාන ද්විත්වයම අසමාන වන අවස්ථාවේදී ප්‍රතිදානය අසත්‍ය වේ.
(False) OR (True)

In an OR gate, when both inputs are not equal, the output is True.
OR ද්වාරයක් තුළදී ආදාන ද්විත්වයම අසමාන වන අවස්ථාවේදී ප්‍රතිදානය සත්‍ය වේ.
True

So, the answer is 3.
එබැවින් පිළිතුර 3 වේ.

4. Which data type does not belong to domain integration?
වසම් ඒකාබද්ධතාවයට අයත් නොවන දත්ත වර්ගය කුමක්ද?

1) Notnull 2) Default 3) Create 4) Check 5) Unique

Answer : 3)

Explanation / පැහැදිලි කිරීම
Domain integrity constraints ensure that the values stored in a database column are valid and follow specific rules. Common domain integrity constraints include.
දත්ත සමුදා තිරුවක ගබඩා කර ඇති අගයන් වලංගු බවත් නිශ්චිත නීති අනුගමනය කරන බවත් වසම් ඒකාබද්ධතා සීමාවන් සහතික කරයි. පොදු වසම් ඒකාබද්ධතා සීමාවන් ඇතුළත් වේ.

NOT NULL
Ensures that a column cannot have NULL values.
තිරුවකට NULL අගයන් තිබිය නොහැකි බව සහතික කරයි.

DEFAULT
Assigns a default value to a column when no value is provided.
අගයක් සපයා නොමැති විට තිරුවකට පෙරනිමි අගයක් පවරයි.

CHECK
Enforces specific conditions on the values entered in a column.
තිරුවක ඇතුළත් කර ඇති අගයන් සඳහා නිශ්චිත කොන්දේසි බලාත්මක කරයි.

UNIQUE
Ensures that all values in a column are unique.
තිරුවක ඇති සියලුම අගයන් අනන්‍ය බව සහතික කරයි.

However, CREATE is not a domain integrity constraint. It is an SQL command used to create database objects like tables, indexes, and views.
කෙසේ වෙතත්, CREATE යනු වසම් ඒකාබද්ධතා සීමාවක් නොවේ. එය වගු, දර්ශක සහ දර්ශන වැනි දත්ත සමුදා වස්තු නිර්මාණය කිරීමට භාවිතා කරන SQL විධානයකි.

Conclusion / නිගමනය
Since CREATE is a DDL (Data Definition Language) command and not a domain integrity constraint, the correct answer is option 3 (CREATE).
CREATE යනු DDL (දත්ත අර්ථ දැක්වීමේ භාෂාව) විධානයක් වන අතර වසම් ඒකාබද්ධතා සීමාවක් නොවන බැවින්, නිවැරදි පිළිතුර විකල්ප 3 (CREATE) වේ.

5. `Employee ( EmpID , EmpName , Dept , Salary , JoinDate )`

Which SQL query correctly retrieves the employee ID, name, and department of employees who have a salary greater than 50,000, sorted in ascending order of department?
දෙපාර්තමේන්තු ආරෝහණ අනුපිළිවෙලට වර්ග කර ඇති, 50,000 ට වැඩි වැටුපක් ඇති සේවකයින්ගේ සේවක හැඳුනුම්පත, නම සහ දෙපාර්තමේන්තුව නිවැරදිව ලබා ගන්නේ කුමන SQL විමසුම මගින්ද?

1)
```
SELECT EmpID, EmpName
FROM Employee
WHERE Salary;
```

2)
```
SELECT EmpID, EmpName, Dept
WHERE Salary > 50000
ORDER BY Dept;
```

3)
```
SELECT EmpID, EmpName, Dept
FROM Employee
WHERE Salary > 50000
ORDER BY Dept ASC;
```

4)
```
SELECT EmpID, Dept
FROM EmpName
WHERE Salary > 50000
ORDER BY Dept ASC;
```

5)
```
SELECT EmpID, EmpName, Dept
FROM Employee;
```

Answer: 3)

Explanation / පැහැදිලි කිරීම:

• **SELECT EmpID, EmpName, Dept**
Retrieves only the required columns: EmpID, EmpName, and Dept.
අවශ්‍ය තීරු පමණක් ලබා ගනී: EmpID, EmpName සහ Dept.

• **FROM Employee**
Specifies the correct table (Employee).
නිවැරදි වගුව (Employee) සඳහන් කරයි.

• **WHERE Salary > 50000**
Filters only employees with a salary greater than 50000.
50000 ට වැඩි වැටුපක් ඇති සේවකයින් පමණක් පෙරහන් කරයි.

• **ORDER BY Dept ASC**
Sorts the results in ascending order (ASC) based on the Dept column.
Dept තීරුව මත පදනම්ව ආරෝහණ අනුපිළිවෙලින් (ASC) ප්‍රතිඵල වර්ග කරයි.

Why are the other options incorrect / අනෙකුත් විකල්ප වැරදි වන්නේ ඇයි?

Option / විකල්ප 1)
Missing a proper comparison condition (WHERE Salary; is incorrect). Should be WHERE Salary > 50000.
නිසි සංසන්දනාත්මක තත්වයක් අස්ථානගත වී ඇත (WHERE Salary; වැරදියි). WHERE Salary > 50000 විය යුතුය.

Option / විකල්ප 2)
Missing the FROM Employee clause.
FROM Employee වගන්තිය අස්ථානගත වී ඇත.

Option / විකල්ප 4)
Incorrect table name (FROM EmpName instead of FROM Employee). Missing EmpName in SELECT clause.
වැරදි වගු නම (FROM EmpName වෙනුවට FROM Employee). SELECT වගන්තියේ EmpName අස්ථානගත වී ඇත.

Option / විකල්ප 5)
No WHERE condition, so it retrieves all employees instead of filtering those with Salary > 50000. No ORDER BY clause, so results are not sorted by Dept.
WHERE කොන්දේසියක් නැත, එබැවින් එය Salary > 50000 ට වැඩි අය filter කිරීම වෙනුවට සියලුම සේවකයින් ලබා ගනී. ORDER BY වගන්තියක් නැත, එබැවින් ප්‍රතිඵල Dept විසින් වර්ග නොකෙරේ.

6. What will be the output of the following Python code?
පහත පයිතන් කේතයේ ප්‍රතිදානය කුමක් වේ ද?

```
def func(a, b=5, c=10):
    return a + b + c
print(func(3, c=3))
```

1) 11   2) 16   3) 21   4) 18   5) error

Answer : 1)

The output of the code is 11. The function func is defined to add three values a, b and c where b and c have default values of 5 and 10, respectively. When calling func(3, c=3), you provide 3 for a and explicitly set c to 3, while b remains at its default value of 5. The function then computes the sum of these values 3 (a) + 5 (b) + 3 (c), which equals 11. Therefore, print(func(3, c=3)) outputs 11.
කේතයේ ප්‍රතිදානය 11 වේ. func ශ්‍රිතය අගයන් තුනක් එකතු කිරීමට අර්ථ දක්වා ඇත a,b සහ c එහිදී b සහ c පිළිවෙළින් 5 සහ 10 හි පෙරනිමි අගයන් ඇත. func(3, c=3) ඇමතීමේ දී, a සඳහා 3 ලබා දෙන අතර c හි අගය 3 ලෙස සකසයි, b හි පෙරනිමි අගය වන 5 එම ලෙසින්ම පවතිනු ඇත. ශ්‍රිතය පසුව මෙම අගයන්හි එකතුව ගණනය කරයි 3 (a) + 5 (b) + 3 (c), එය 11 ට සමාන වේ. එබැවින්, print(func(3, c=3)) 11.

7. What will be the output of the following Python code?
පහත පයිතන් කේතයේ ප්‍රතිදානය කුමක් වේද?

```
numbers = [1, 2, 3, 4, 5]
squares = [n**2 for n in numbers if n % 2 == 0]
print(squares)
```

1) [1, 4, 9, 16, 25] 2) [1, 4, 9, 16] 3) [4, 16]
4) [1, 9, 25] 5) [4, 9, 16, 25]

Answer : 3)

Explanation / පැහැදිලි කිරීම

numbers = [1, 2, 3, 4, 5]
This line creates a list called numbers containing the integers [1, 2, 3, 4, 5].
මෙම රේඛාව නිඛිල [1, 2, 3, 4, 5] අඩංගු numbers නම් ලැයිස්තුවක් නිර්මාණය කරයි.

squares = [n2 for n in numbers if n % 2 == 0]**
This is a list comprehension that creates a new list called squares.
මෙය squares නම් අලුත් ලයිස්තුවක් නිර්මාණය කරන list comprehension එකකි.

for loop iterates over each element n in the numbers list. if n % 2 == 0 checks whether the number n is even (when divided by 2, it leaves a remainder of 0). For each even number, it calculates the square of n (n**2) and includes it in the squares list.
for ලූපය සඳහා numbers ලැයිස්තුවේ එක් එක් මූලාංග n මත පුනරාවර්තනය වේ. if n % 2 == 0 අංකය ඉරට්ටේ දැයි පරීක්ෂා කරයි (2 න් බෙදූ විට, එයට 0 ක් ඉතිරි වේ). එක් එක් ඉරට්ටේ සංඛ්‍යාව සඳහා, එය n හි වර්ගය (n**2) ගණනය කර එය squares ලැයිස්තුවට ඇතුළත් කරයි.

print(squares)
This line prints the list squares.
මෙම රේඛාව squares ලැයිස්තුව මුද්‍රණය කරයි.

From the numbers list [1, 2, 3, 4, 5], the even numbers are 2 and 4.
[1, 2, 3, 4, 5] ඇතුළත් numbers ලැයිස්තුවෙන් ඉරට්ටේ අංක 2 සහ 4 වේ.

- The square of 2 is 4 (2**2 = 4).
  2 හි වර්ගය 4 වේ. (2**2 = 4)
- The square of 4 is 16 (4**2 = 16).
  4 හි වර්ගය 16 වේ. (4**2 = 16)

Therefore, the output is 3) [4, 16].
එබැවින්, ප්‍රතිදානය 3) [4, 16] වේ.

8. Correct statement about python is,
පයිතන් පිළිබඳව නිවැරදි ප්‍රකාශය වන්නේ,

1) Python uses semicolons at the end of each statement.
පයිතන් එක් එක් ප්‍රකාශය අවසානයේ semicolan භාවිතා කරයි.

2) Python is strictly a compiled language.
පයිතන් යනු දැඩි ලෙස සම්පාදිත භාෂාවකි.

3) The `range()` function is memory efficient than lists.
`range()` ශ්‍රිතය ලැයිස්තු වලට වඩා මතක කාර්යක්ෂම වේ.

4) Python's `pass` statement is an operational statement.
පයිතන් හි `pass` ප්‍රකාශය මෙහෙයුම් ප්‍රකාශයකි.

5) Python has a built-in `switch` statement.
`switch` ප්‍රකාශයක් පයිතන් හි අන්තර්ගතව ඇත.

Answer: 3)

python doesn't use semicolons and python is both compiled and interpreted so it is not a strictly compiled language, python pass statement is not an operational statement and no python doesn't have a built in switch statement, range() function is memory efficient than a list because range() function only stores the starting value end value and the step value but list stores all the elements in it.
පයිතන් තිත් කොමාව භාවිතා නොකරන අතර python compiling සහ interpreting දෙකම භාවිතා කරයි, එබැවින් එය දැඩි ලෙස සම්පාදිත කරන ලද භාෂාවක් නොවේ, python pass ප්‍රකාශය මෙහෙයුම් ප්‍රකාශයක් නොවේ. python සතුව ස්විච් ප්‍රකාශයක් ගොඩනගා නැත, range() ශ්‍රිතය ලැයිස්තුවකට වඩා මතකය කාර්යක්ෂම වේ මන්ද range() ශ්‍රිතය ගබඩා කරන්නේ ආරම්භක අගය අවසාන අගය සහ පියවර අගය පමණක් වන නමුත් ලැයිස්තුව එහි ඇති සියලුම මූලාංග ගබඩා කරයි.

9. What is the output of the following Python code if the input is given as 2, 4, 6, 8, 10, 12, 14, 16, 18 and 20 respectively?
ආදානය පිළිවෙලින් 2, 4, 6, 8, 10, 12, 14, 16, 18 සහ 20 ලෙස ලබා දෙන්නේ නම් පහත පයිතන් කේතයේ ප්‍රතිදානය කුමක්ද?

```
initial = 0
for k in range (10) :
      x = input ('Enter Values : ') ; x = int (x)
      initial = initial + x
print (initial)
```

1) `55`   2) `100`   3) `110`   4) `25`   5) `error`

Answer : 3)

Let's analyze the python code.
පයිතන් කේතය විශ්ලේෂණය කර බලමු.

- The code initializes a variable **`initial`** to 0.
කේතය **`initial`** විචල්‍ය 0 සිට ආරම්භ කරයි.

- A for loop runs 10 times, corresponding to the **`range(10)`** function.
for ලූපය 10 වරක් ධාවනය වන අතර, **`range(10)`** ශ්‍රිතයට අනුරූප වේ.

- In each iteration, the program prompts the user to input a value, converts that input to an integer, and adds it to the initial variable.
එක් එක් පුනරාවර්තනයකදී, වැඩසටහන මගින් අගයක් ඇතුළත් කිරීමට පරිශීලකයා පොළඹවයි, එම ආදානය පූර්ණ සංඛ්‍යාවකට පරිවර්තනය කරයි, සහ එය ආරම්භක විචල්‍යයට එක් කරයි.

After all 10 inputs are processed, the final value of initial is printed.
සියලුම ආදාන 10 ක් සැකසීමෙන් පසු, initial හි අවසාන අගය මුද්‍රණය කෙරේ.

- The inputs provided are **2, 4, 6, 8, 10, 12, 14, 16, 18, 20**.
සපයා ඇති ආදාන **2, 4, 6, 8, 10, 12, 14, 16, 18, 20** වේ.

- The sum of these numbers is calculated as,
  මෙම සංඛ්‍යාවල එකතුව ගණනය කරනු ලබන්නේ,

  2 + 4 + 6 + 8 + 10 + 12 + 14 + 16 + 18 + 20 = 110.

Therefore, the output of the code will be 110.
එබැවින්, කේතයේ ප්‍රතිදානය 110 වනු ඇත.

So, the correct answer is 3).
ඒබැවින්, නිවැරදි පිළිතුර 3) වේ.

10. You want to make an image responsive so that it scales with the screen size. Which of the following approaches is correct?
ඔබට image එකක් responsive කිරීමට අවශ්‍ය වන අතර එය තිරයේ ප්‍රමාණය සමඟ වෙනස් වේ. පහත සඳහන් ප්‍රවේශයන්ගෙන් නිවැරදි වන්නේ කුමක්ද?

1) <img src="image.jpg" style="width: 100%; height: auto;">
2) <img src="image.jpg" style="width: 500px; height: 300px;">
3) <img src="image.jpg" style="width: auto; height: auto;">
4) <img src="image.jpg" style="width: 100%; height: 100%;">
5) <img src="image.jpg" style="width: 100px; height: 100px;">

Answer : 1)

Explanation / පැහැදිලි කිරීම:

To make an image **responsive**, it should **scale with the screen size** while maintaining its aspect ratio.
image එකක් **responsive** කිරීමට, එහි දර්ශන අනුපාතය පවත්වා ගනිමින් එය **screen size සමඟ පරිමාණය** කළ යුතුය.

- **width: 100%** → Makes the image take the full width of its container. රූපය එහි container එකෙන් සම්පූර්ණ පළල ගැනීමට සලස්වයි
- **height: auto** → Automatically adjusts the height to maintain the aspect ratio. aspect ratio එක පවත්වා ගැනීමට උස ස්වයංක්‍රියව සීරුමාරු කරයි.

Why the other options are incorrect? / වෙනත් විකල්ප වැරදි ඇයි ?

2) width: 500px; height: 300px;
Fixed width and height **do not scale** with the screen size, making it non-responsive.
ස්ථාවර පළල සහ උස තිරයේ ප්‍රමාණය සමඟ පරිමාණය නොකරන අතර එය **ප්‍රතිචාර නොදක්වයි.**

3) width: auto; height: auto;
This keeps the **image at its original size**, which **does not** ensure responsiveness.
මෙමඟින් රූපය එහි **මුල් ප්‍රමාණයේ** තබා ගන්නා අතර එමඟින් ප්‍රතිචාර දැක්වීම සහතික නොවේ.

4) width: 100%; height: 100%;
This **distorts** the image by stretching it instead of maintaining its aspect ratio.
මෙය එහි දර්ශන අනුපාතය පවත්වා ගැනීම වෙනුවට එය දිගු කිරීමෙන් රූපය **විකෘති** කරයි.

5) width: 100px; height: 100px;
Fixed dimensions prevent the image from resizing dynamically.
ස්ථාවර මානයන් රූපය ගතිකව ප්‍රතිප්‍රමාණය වීම වළක්වයි.