සියලු ම හිමිකම් ඇවිරිණි /All Rights Reserved]

Ravindu Bandaranayake - ictfromabc

Information and Communication Technology
තොරතුරු හා සන්නිවේදන තාක්ෂණය

2026 QUIZ NO 380 MARKING

2025/12/25
ictfromabc
Ravindu Bandaranayake

1. Which of the following describes a characteristic of valuable information?
පහත සඳහන් දේවලින් වටිනා තොරතුරුවල ලක්ෂණයක් විස්තර කරන්නේ කුමක්ද?

1) Requires the least amount of data
අවම දත්ත ප්‍රමාණයක් අවශ්‍ය වේ.

2) Provides all the available data
පවතින සියලුම දත්ත සපයයි.

3) Presented within a meaningful context
අර්ථවත් සන්දර්භයක් තුළ ඉදිරිපත් කර ඇත.

4) Is complex enough to require expert analysis
විශේෂඥ විශ්ලේෂණයක් අවශ්‍ය තරමට සංකීර්ණ වේ.

5) Must be digital to be valuable
වටිනා වීමට අංකිත විය යුතුය.

Answer : 3)

Let's go through the given options to find the statement that describes a characteristic of valuable information
වටිනා තොරතුරු වල ලක්ෂණයක් විස්තර කරන ප්‍රකාශය සොයා ගැනීමට ලබා දී ඇති විකල්ප හරහා යමු

1) Requires the least amount of data/ අවම දත්ත ප්‍රමාණයක් අවශ්‍ය වේ
Valuable information doesn't necessarily mean less data; it is more about how useful the data is in the given context.
වටිනා තොරතුරු යනු අඩු දත්ත ප්‍රමාණයක් නොවේ. එය ලබා දී ඇති සන්දර්භය තුළ දත්ත කෙතරම් ප්‍රයෝජනවත්ද යන්න වඩා වැදගත් වේ.

2) Provides all the available data/ පවතින සියලුම දත්ත සපයයි
Providing all data might not always be valuable. It can confuse the user the user if it's not relevant or organized effectively.
සියලුම දත්ත සැපයීම සැමවිටම වටිනා නොවිය හැක. එය අදාළ නොවේ නම් හෝ ඵලදායී ලෙස සංවිධානය වී ඇත්නම් එය පරිශීලකයා ව්‍යාකූල කිරීමට හැකිය.

3) Presented within a meaningful context/ අර්ථවත් සන්දර්භයක් තුළ ඉදිරිපත් කර ඇත
Information becomes valuable when it is presented in a context that helps make sense of it. Data alone is not as useful unless interpreted properly.
තොරතුරු වැදගත් වන්නේ එය අර්ථවත් කිරීමට උපකාර වන සන්දර්භයක් තුළ ඉදිරිපත් කරන විටය. නිවැරදිව අර්ථකථනය නොකළහොත් දත්ත පමණක් එතරම් ප්‍රයෝජනවත් නොවේ.

4) Is complex enough to require expert analysis/ විශේෂඥ විශ්ලේෂණයක් අවශ්‍ය වන තරමට සංකීර්ණ වේ
Complexity does not always equate to value. Valuable information can be simple as long as it provides the insights needed.
සංකීර්ණත්වය සැමවිටම වටිනාකමට සමාන නොවේ. සරල තොරතුරු මගින් අවශ්‍ය තීක්ෂ්ණ බුද්ධිය ලබාදීම සිදුවිය හැක.

5) Must be digital to be valuable/ වටිනා වීමට අංකිත විය යුතුය
Information does not have to be digital to be valuable. Valuable information can be in any format, as long as it serves its purpose effectively.
තොරතුරු වටිනා වීමට අංකිත විය යුතු නැත. වටිනා තොරතුරු එහි අරමුණ ඵලදායී ලෙස ඉටු කරන තාක් කල්, ඕනෑම ආකෘතියකින් තිබිය හැකිය.

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

2. Convert the decimal number 156.75 to binary.
දශමය 156.75 ද්විමය බවට පරිවර්තනය කරන්න.

1) $\underline{10011100.11_2}$ 2) 10011101.01_2 3) 10011100.101_2 4) 10011101.10_2 5) 10011100.111_2

Answer : 1)

To convert the decimal number 156.75 to binary, we need to convert both the integer part (156) and the fractional part 0.75 separately.
දශමය සංඛ්‍යාව 156.75 ද්විමය බවට පරිවර්තනය කිරීම සඳහා, අපි පූර්ණ සංඛ්‍යා කොටස 156 සහ දශම කොටස 0.75 යන දෙකම වෙන වෙනම පරිවර්තනය කළ යුතුය.

1. Converting the integer part (156):
පූර්ණ සංඛ්‍යා කොටස පරිවර්තනය කිරීම (156)

Divide the number by 2 and write the remainder.
අංකය 2න් බෙදලා ඉතිරිය ලියාගමු.

Let's keep dividing by 2 until we reach 0.
0 ට ළඟා වන තුරු 2 න් බෙදීම දිගටම කරගෙන යමු.

156
78 – 0
39 – 0
19 – 1
9 – 1
4 – 1
2 – 0
1 – 0
0 – 1

10011100_2

2. Converting the decimal part (0.75):
දශම කොටස පරිවර්තනය කිරීම (0'75)

Multiply the fraction by 2 and note the whole number part.
භාගය 2න් ගුණ කර පූර්ණ සංඛ්‍යා කොටස සටහන් කරගනී.

Keep multiplying the fraction by 2 until you reach 0 or a repeating fraction.
ඔබ 0 හෝ පුනරාවර්තන භාගයකට ළඟා වන තෙක් භාගික කොටස 2 න් ගුණ කිරීම දිගටම කරගෙන යමු.

	0.75 x 2
1	.50 x 2
1	.00 x 2
0	.00

.11

Accordingly, the correct answer is, 10011100.11_2
ඒ අනුව නිවැරදි පිළිතුර වන්නේ, 10011100.11_2

3. Which of the following Boolean expressions can be implemented using only one NAND gate, when simplified?
සරල කළ විට, එක් NAND ද්වාරයක් පමණක් භාවිතයෙන් ක්‍රියාත්මක කළ හැක්කේ පහත සඳහන් බූලියන් ප්‍රකාශනවලින් කවරක්ද?

1) $C\overline{D} + 0(A\overline{A} + \overline{BD})$
2) $C1 + A(\overline{AB} + \overline{C}1)$
3) $\underline{\overline{1(AB + CB)} + \overline{AB}}$
4) $BC + C(A\overline{B} + BC)$
5) $(A \oplus B)C + ABC$

Answer : 3)

Let's simplify the following expressions to determine which one can be implemented using a NAND gate.
NAND ද්වාරයක් භාවිතයෙන් ක්‍රියාත්මක කළ හැක්කේ කුමක්ද යන්න තිරණය කිරීමට පහත ප්‍රකාශන සරල කරමු.

1) $\mathbf{C\overline{D} + 0(A\overline{A} + \overline{BD})}$

$C\overline{D} + 0(A\overline{A} + \overline{BD})$	
$C\overline{D} + 0(A\overline{A} + \overline{B} + \overline{D})$	De morgan's Law ද මුවාර් ප්‍රමේයය
$C\overline{D} + 0$	Identity Law සර්වසාම්‍ය න්‍යාය
$C\overline{D}$	Identity Law සර්වසාම්‍ය න්‍යාය

2) $C1 + A(\overline{AB} + \overline{C}1)$

$C1 + A(\overline{AB} + \overline{C}1)$	
$C1 + A(\overline{A} + \overline{B} + \overline{C}1)$	De morgan's Law ද මුවාර් ප්‍රමේයය
$C + A(\overline{A} + \overline{B} + \overline{C}1)$	Identity Law සර්වසාම්‍ය න්‍යාය
$C + A(\overline{A} + \overline{B} + \overline{C})$	Identity Law සර්වසාම්‍ය න්‍යාය
$C + A\overline{A} + A\overline{B} + A\overline{C}$	Distributive Law විඝටන න්‍යාය
$C + 0 + A\overline{B} + A\overline{C}$	Complement Law ප්‍රතිලෝම න්‍යාය
$C + A\overline{B} + A\overline{C}$	Identity Law සර්වසාම්‍ය න්‍යාය
$C + A\overline{B} + A$	Redundancy Law සමතිරික්ත න්‍යාය
$C + A$	Redundancy Law සමතිරික්ත න්‍යාය

3) $\mathbf{\overline{1(AB + CB)} + \overline{AB}}$

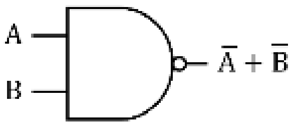
$\overline{1(AB + CB)} + \overline{AB}$	
$\overline{1} + \overline{(AB + CB)} + \overline{AB}$	De morgan Law ද මුවාර් ප්‍රමේයය
$\overline{1} + (\overline{A} + \overline{B})(\overline{C} + \overline{B}) + \overline{AB}$	De morgan Law ද මුවාර් ප්‍රමේයය
$\overline{1} + (\overline{A} + \overline{B})(\overline{C} + \overline{B}) + \overline{A} + \overline{B}$	De morgan Law ද මුවාර් ප්‍රමේයය
$0 + (\overline{A} + \overline{B})(\overline{C} + \overline{B}) + \overline{A} + \overline{B}$	$\overline{1} = 0$
$(\overline{A} + \overline{B})(\overline{C} + \overline{B}) + \overline{A} + \overline{B}$	Identity Law සර්වසාම්‍ය න්‍යාය
$(\overline{C} + \overline{B})\overline{A}(\overline{C} + \overline{B})\overline{B} + \overline{A} + \overline{B}$	Distributive Law විඝටන න්‍යාය
$(\overline{C} + \overline{B})\overline{B} + \overline{A} + \overline{B}$	Redundancy Law සමතිරික්ත න්‍යාය
$\overline{A} + \overline{B}$	Redundancy Law සමතිරික්ත න්‍යාය

4) $\mathbf{BC + C(A\overline{B} + BC)}$

$BC + C(A\overline{B} + BC)$	
$BC + CA\overline{B} + CBC$	Distributive Law විඝටන න්‍යාය
$BC + CA\overline{B} + CB$	Idempotent Law තද්වභාවී න්‍යාය
$BC + CA\overline{B}$	Idempotent Law තද්වභාවී න්‍යාය
$C(A\overline{B} + B)$	Distributive Law විඝටන න්‍යාය
$C(A + B)$	Redundancy Law සමතිරික්ත න්‍යාය
$CA + CB$	Distributive Law විඝටන න්‍යාය

5) $(A \oplus B)C + ABC$

$(A \oplus B)C + ABC$	
$(A.\overline{B}) + (\overline{A}.B)C + ABC$	
$(A.\overline{B}.C) + (\overline{A}.B.C) + ABC$	Distributive Law විඝටන න්‍යාය
$C(A.\overline{B} + AB) + (\overline{A}.B.C)$	Distributive Law විඝටන න්‍යාය
$CA + (\overline{A}.B.C)$	Redundancy Law සමතිරික්ත න්‍යාය
$C(A + \overline{A}.B)$	Distributive Law විඝටන න්‍යාය
$C(A + B)$	Redundancy Law සමතිරික්ත න්‍යාය



So, according to the above explanation the answer is 3).
එබැවින්, ඉහත පැහැදිලි කිරීම අනුව පිළිතුර 3) වේ.

4. A developer wants to create a database called **"pcGames"**. Which of the following SQL statements should be used to achieve this task?
සංවර්ධකයෙකුට **"pcGames"** නමින් දත්ත සමුදායක් නිර්මාණය කිරීමට අවශ්‍යයි. මෙම කාර්යය සාක්ෂාත් කර ගැනීම සඳහා පහත සඳහන් SQL ප්‍රකාශවලින් කුමන ප්‍රකාශයක් භාවිතා කළ යුතුද?

   1) `BUILD DATABASE pcGames;`
   2) `NEW SCHEMA pcGames;`
   3) `CONNECT TO pcGames;`
   4) `CREATE TABLE pcGames;`
   5) `CREATE DATABASE pcGames;`

   Answer : 5)

   Explanation / පැහැදිලි කිරීම

   To create a new database called `"pcGames"`, the correct SQL statement is `CREATE DATABASE pcGames;`, which is the standard command supported by most relational databases like MySQL, PostgreSQL, and SQL Server. The other options are incorrect: Option 1 (`BUILD DATABASE`) is not a valid SQL command, Option 2 (`NEW SCHEMA`) only creates a namespace but not a database, Option 3 (`CONNECT TO`) connects to an existing database without creating it, and Option 4 (`CREATE TABLE`) only creates a table rather than a database.
   `"pcGames"`නමින් නව දත්ත සමුදායක් නිර්මාණය කිරීම සඳහා, නිවැරදි SQL ප්‍රකාශය වන්නේ (`CREATE DATABASE pcGames;` වේ. එය MySQL, PostgreSQL සහ SQL Server වැනි බොහෝ සම්බන්ධිත දත්ත සමුදායන් විසින් සහාය දක්වන සම්මත විධානයයි. අනෙක් විකල්ප වැරදියි: විකල්ප (`BUILD DATABASE`) වලංගු SQL විධානයක් නොවේ, විකල්ප 2 (`NEW SCHEMA`) නාම අවකාශයක් පමණක් නිර්මාණය කරයි නමුත් දත්ත සමුදායක් නොවේ, විකල්ප 3 (`CONNECT TO`) එය නිර්මාණය නොකර පවතින දත්ත ගබඩාවකට සම්බන්ධ කරයි, සහ විකල්ප 4 (`CREATE TABLE`) දත්ත සමුදායක් වෙනුවට වගුවක් පමණක් නිර්මාණය කරයි.

   Therefore, Option 5 is the only valid and appropriate choice.
   එබැවින්, විකල්ප 5 එකම වලංගු සහ සුදුසු තේරීම වේ.

5. Which of the following is NOT a type of JOIN operation in SQL?
පහත සඳහන් ඒවායින් SQL හි JOIN මෙහෙයුම් වර්ගයක් නොවන්නේ කුමක්ද?

   1) INNER JOIN 2) OUTER JOIN 3) CROSS JOIN 4) LEFT JOIN 5) BACK JOIN

   Answer : 5)

   Explanation / පැහැදිලි කිරීම

   BACK JOIN is not a recognized or standard type of JOIN operation in SQL. The other options are valid types of JOIN operations commonly used in SQL.

BACK JOIN යනු SQL හි පිළිගත් හෝ සම්මත ආකාරයේ JOIN මෙහෙයුමක් නොවේ. අනෙකුත් විකල්ප SQL හි බහුලව භාවිතා වන වලංගු JOIN මෙහෙයුම් වේ.

- **INNER JOIN**:
Returns rows when there is a match in both tables.
වගු දෙකෙහිම ගැලපීමක් ඇති විට පේළි ලබා දෙයි.

- **OUTER JOIN**:
A general term that can include LEFT JOIN, RIGHT JOIN, and FULL OUTER JOIN, depending on how unmatched rows are handled.
නොගැලපෙන පේළි හසුරුවන ආකාරය මත පදනම්ව, වම් එක්වීම, දකුණට එකතු වීම සහ සම්පූර්ණ පිටත එකතු කිරීම ඇතුළත් විය හැකි සාමාන්‍ය පදයකි.

- **CROSS JOIN**:
Produces the Cartesian product of the two tables (all possible combinations of rows).
වගු දෙකේ කාටිසියානු නිෂ්පාදිතය (පේළිවල හැකි සියලුම සංයෝජන) නිෂ්පාදනය කරයි.

LEFT JOIN:
Returns all rows from the left table, and the matching rows from the right table (or NULL if there is no match).
වම් වගුවෙන් සියලුම පේළි සහ දකුණු වගුවෙන් ගැලපෙන පේළි ආපසු ලබා දෙයි (හෝ ගැළපීමක් නොමැති නම් NULL

Therefore the Correct answer is 5)
එබැවින් නිවැරදි පිළිතුර 5)

6. Below is a table 'student' with details of students of a school. What is the correct SQL code to change the Kavindu's class to 12A?
පාසලක ශිෂ්‍යයන් පිළිබඳ විස්තර ඇති student වගුව පහත දැක්වේ. එහි කවිඳු ශිෂ්‍යයාගේ පන්තිය 12A ලෙස වෙනස් කිරීමට අදාළ නිවැරදි SQL කේතය කුමක්ද?

student

id	name	gender	telephone	class
001	Ravindu	M	0714555513	12D
002	Kavindu	M	0772222222	12C
003	Tharindu	M	0703333333	12B

1) `UPDATE student ALTER class = '12A' WHERE id = 002`
2) <u>`UPDATE student SET class = '12A' WHERE id = 002`</u>
3) `INSERT INTO student ALTER class = '12A' WHERE id = 002`
4) `UPDATE student SET class = '12C' WHERE name='Kavindu'`
5) `UPDATE student SELECT class = '12A' WHERE id = 002`

Answer : 2)

UPDATE: This is the correct SQL command to modify existing records in a table. (in this case, student)
UPDATE: වගුවක (මෙම අවස්ථාවේදී, student) පවතින වාර්තා වෙනස් කිරීම සඳහා නිවැරදි SQL විධානය මෙයයි.

SET: Specifies the column to be updated (in this case, class=' 12A') and the new value to assign.
SET: යාවත්කාලීන කළ යුතු තීරුව (මෙම අවස්ථාවේදී, class=' 12A') සහ පැවරිය යුතු නව අගය නියම කරයි.

WHERE: Specifies the condition that identifies the record(s) to be updated (here, id = 002).
WHERE: යාවත්කාලීන කළ යුතු වාර්තා(ව) හඳුනා ගන්නා කොන්දේසිය නියම කරයි (මෙහි, id = 002)

```
UPDATE student SET class = '12A' WHERE id = 002
```

So, the correct answer number is 2.
එමනිසා, නිවැරදි පිළිතුරු අංකය 2 වේ.

7. Which of the following statements is true about Python tuples?
Python tuples සම්බන්ධයෙන් පහත සඳහන් කුමන ප්‍රකාශය සත්‍යද?

1) Tuples are mutable. Tuples වෙනස් කල හැක.
2) Tuples are unordered. Tuples අනුපිලිවෙලක් නැත.
3) Tuples can contain duplicate elements. Tuples වල අනුපිටපත් මූලාංග අඩංගු විය හැක.
4) Tuples can grow in size. Tuplesවල ප්‍රමාණය වැඩි කළ හැක.
5) None of the above. ඉහත කිසිවක් නොවේ.

Answer : 3)

Let's review each option.
අපි එක් එක් විකල්පය සමාලෝචනය කරමු.

1) Tuples are mutable / Tuples වෙනස් කල හැක
This statement is false. Tuples in Python are immutable, meaning their contents cannot be changed once they are created. You cannot add, remove, or modify elements in a tuple.
මෙම ප්‍රකාශය අසත්‍ය වේ. Python හි Tuples වෙනස් කළ නොහැකි ය, එනම් ඒවා නිර්මාණය කළ පසු ඒවායේ අන්තර්ගතය වෙනස් කළ නොහැක. ඔබට ටපල් එකක මූලද්‍රව්‍ය එකතු කිරීමට, ඉවත් කිරීමට හෝ වෙනස් කිරීමට නොහැක.

2) Tuples are unordered / Tuples අනුපිලිවෙලක් නැත
This statement is false. Tuples in Python are ordered collections, meaning the elements have a defined order, and this order will not change.
මෙම ප්‍රකාශය අසත්‍ය වේ. Python හි Tuples යනු ඇණවුම් කළ එකතුවකි, එනම් මූලද්‍රව්‍ය වලට නිශ්චිත අනුපිළිවෙලක් ඇති අතර මෙම අනුපිළිවෙල වෙනස් නොවේ.

3) Tuples can contain duplicate elements / Tuples වල අනුපිටපත් මූලද්‍රව්‍ය අඩංගු විය හැක
This is the true statement. Tuples can contain duplicate elements, meaning the same value can appear more than once in a tuple.
මෙය සත්‍ය ප්‍රකාශය වේ. Tuples හි අනුපිටපත් මූලද්‍රව්‍ය අඩංගු විය හැක, එනම් tuple එකක එකම අගය එක වරකට වඩා දිස්විය හැක.

4) Tuples can grow in size / Tuples වල ප්‍රමාණය වැඩි කල හැක
This statement is false. Tuples are immutable, so once a tuple is created, its size cannot be changed. You cannot add or remove elements from a tuple.
මෙම ප්‍රකාශය අසත්‍ය වේ. Tuples වෙනස් කළ නොහැක, එබැවින් tuple නිර්මාණය කළ පසු එහි විශාලත්වය වෙනස් කළ නොහැක. ඔබට ටපල් එකකින් මූලද්‍රව්‍ය එකතු කිරීමට හෝ ඉවත් කිරීමට නොහැක.

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

8. What is the output of the following code?
පහත කේතයේ ප්‍රතිදානය කුමක්ද?

```
data = [15, 22, 8, 19, 31, 4]
i = 0
result = []
while i < len(data):
    if data[i] % 2 == 0:
        i += 1
        continue
    result.append(data[i] * 2)
    i += 1
print(result)
```

1) [30, 38, 62]
2) [38, 62]
3) [15, 8, 19, 31]
4) [30, 38, 62, 8]
5) [15, 19, 31]

Answer : 1)

Step-by-Step Execution / පියවරෙන් පියවර ක්‍රියාත්මක කිරීම

1. Initial Setup / මූලික සැකසුම

- `data: [15, 22, 8, 19, 31, 4]`
- `i: 0` (starting index).
- `result: []` (empty list).

2. Iterations in the while loop / while ලූපයේ පුනරාවර්තන

Step	Index (i)	Value (data[i])	Condition (data[i] % 2 == 0)	Action	Updated result	Next i
1	0	15	False (odd)	Append 15 * 2 = 30	[30]	1
2	1	22	True (even)	Skip (continue)	[30]	2
3	2	8	True (even)	Skip (continue)	[30]	3
4	3	19	False (odd)	Append 19 * 2 = 38	[30, 38]	4
5	4	31	False (odd)	Append 31 * 2 = 62	[30, 38, 62]	5
6	5	4	True (even)	Skip (continue)	[30, 38, 62]	6

3. Loop Termination / ලූපය අවසන් කිරීම

- The while loop ends when i = 6 (index exceeds the length of data).
  i = 6 (දර්ශකය දත්ත දිග ඉක්මවන විට) විට ලූපය අවසන් වේ.

Final Result / අවසාන ප්‍රතිඵලය

- **`result`**`: [30, 38, 62]`

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

```
def find_max(nums):
    max_val = nums[0]
    for n in nums:
        if n > max_val:
            max_val = n
    return max_val

vals = [-3, -1, -4, -2, -5]
print(find_max(vals))
```

1) `-1` 2) `1` 3) `-2` 4) `2` 5) `0`

Answer : 1)

The function `find_max(nums)` finds the maximum value in a list of numbers. It starts by assigning the first element of the list (`` `nums[0]` ``) to `max_val`. Then, it iterates through the list, comparing each number (`` `n` ``) with `max_val`. If `n` is greater than `max_val`, it updates `max_val` to that value. In this case, the list `vals = [-3, -1, -4, -2, -5]` is processed, and since `-1` is the largest number in the list, the function returns `-1`. Therefore, the output is `-1`.
`find_max(nums)` ශ්‍රිතය සංඛ්‍යා ලැයිස්තුවක උපරිම අගය සොයා ගනී. එය ආරම්භ වන්නේ ලැයිස්තුවේ පළමු මූලද්‍රව්‍යය (`` `nums[0]` ``) `max_val` වෙත පැවරීමෙනි. ඉන්පසුව, එය එක් එක් සංඛ්‍යාව (`` `n` ``) `max_val` සමඟ සංසන්දනය කරමින් ලැයිස්තුව හරහා පුනරාවර්තනය වේ. `n` `max_val` ට වඩා වැඩි නම්, එය එම අගයට `max_val` යාවත්කාලීන කරයි. මෙම අවස්ථාවෙහිදී, ලැයිස්තුව `vals = [-3, -1, -4, -2, -5]` සකසන ලද අතර, `-1` ලැයිස්තුවේ විශාලතම අංකය වන බැවින්, ශ්‍රිතය `-1` ලබා දෙයි. . එබැවින්, ප්‍රතිදානය `-1` වේ.

10. What is the attribute that cannot be use inside the image tag?
image උසුලනය තුල භාවිතා කළ නොහැකි ගුණාංගය කුමක්ද?

1) src 2) width 3) height 4) alt 5) controls

Answer : 5)

Explanation / පැහැදිලි කිරීම

In HTML, the <img> tag is used to embed images in a webpage. Key attributes of this tag include:
HTML හි, <img> උසුලනය වෙබ් පිටුවක රූප ඇතුළත් කිරීමට භාවිතා කරයි. මෙම උසුලනයේ ප්‍රධාන උපලක්ෂණවලට ඇතුළත් වන්නේ:

Option / විකල්ප 1)
specifies the path of the image file / රූප ගොනුවේ මාර්ගය නියම කරයි,

Option / විකල්ප 2)
defines the width of the image / රූපයේ පළල නිර්වචනය කරයි,

Option / විකල්ප 3)
sets the height of the image / රූපයේ උස සකසයි,

Option / විකල්ප 4)
provides alternative text for the image if it cannot be displayed.
රූපය ප්‍රදර්ශනය කළ නොහැකි නම් එය සඳහා විකල්ප පාඨයක් සපයයි.

Option / විකල්ප 5)
The **controls (5)** attribute, however, is used with media elements like <audio> or <video> to provide playback controls, not with the <img> tag. Therefore, controls is not a valid attribute for an image tag.
කෙසේ වෙතත්, controls **(5)** උපලක්ෂණය <img> උසුලනය සමඟ නොව, නැවත playback එකක් පාලනයන් සැපයීමට <audio> හෝ <video> වැනි මාධ්‍ය අංග සමඟ භාවිතා වේ. එබැවින්, controls යනු රූප උසුලනයක් සඳහා වලංගු ගුණාංගයක් නොවේ.

Therefore the Correct answer is 5)
එබැවින් නිවැරදි පිළිතුර 5)