සියලු ම හිමිකම් ඇවිරිණි /All Rights Reserved]

Ravindu Bandaranayake - ictfromabc

Information and Communication Technology
තොරතුරු හා සන්නිවේදන තාක්ෂණය

2026 QUIZ NO 529 MARKING

2026/05/23
ictfromabc
Ravindu Bandaranayake

1. Which statement(s) is/are incorrect regarding custom-developed software?
අභිරුචි-සංවර්ධිත මෘදුකාංග සම්බන්ධයෙන් කුමන ප්‍රකාශය/ප්‍රකාශ වැරදි ද?

A- Custom software is designed to meet the specific and unique needs of a user or organization.
අභිරුචි මෘදුකාංග නිර්මාණය කර ඇත්තේ පරිශීලකයෙකුගේ හෝ සංවිධානයක නිශ්චිත සහ අද්විතීය අවශ්‍යතා සපුරාලීම සඳහා ය.

B- Custom software is typically cheaper and faster to acquire than COTS software.
අභිරුචි මෘදුකාංග සාමාන්‍යයෙන් COTS මෘදුකාංගවලට වඩා ලාභදායී සහ වේගවත් වේ.

C- Custom software often requires significant development time and resources.
අභිරුචි මෘදුකාංග සඳහා බොහෝ විට සැලකිය යුතු සංවර්ධන කාලයක් සහ සම්පත් අවශ්‍ය වේ.

1) A only A පමණි	2) B only B පමණි	3) C only C පමණි	4) A and B only A හා B පමණි	5) B and C only B හා C පමණි

Answer : 2)

Explanation of each statement / එක් එක් ප්‍රකාශය පැහැදිලි කිරීම:

Statement / ප්‍රකාශය A
This statement is correct. The primary purpose of custom software is to tailor it precisely to the requirements of the user or organization, rather than being a general solution.
මෙම ප්‍රකාශය නිවැරදියි. අභිරුචි මෘදුකාංගවල මූලික අරමුණ වන්නේ එය සාමාන්‍ය විසඳුමක් වීමට වඩා පරිශීලකයාගේ හෝ සංවිධානයේ අවශ්‍යතාවලට හරියටම ගැලපෙන පරිදි සකස් කිරීමයි.

Statement / ප්‍රකාශය B
This statement is incorrect. COTS (Commercial Off-The-Shelf) software is generally cheaper and faster to acquire because it's pre-built and readily available. Custom software involves development from scratch, which incurs higher costs (development, testing, maintenance) and takes more time to build and implement.
මෙම ප්‍රකාශය වැරදියි. COTS මෘදුකාංග සාමාන්‍යයෙන් ලාභදායී සහ අත්පත් කර ගැනීමට වේගවත් වන්නේ එය පෙර-නිර්මාණය කර ඇති සහ පහසුවෙන් ලබා ගත හැකි බැවිනි. අභිරුචි මෘදුකාංග සඳහා මුල සිටම සංවර්ධනය ඇතුළත් වන අතර, එය ඉහළ පිරිවැයක් (සංවර්ධනය, පරීක්ෂා කිරීම, නඩත්තු කිරීම) දරන අතර ගොඩනැගීමට සහ ක්‍රියාත්මක කිරීමට වැඩි කාලයක් ගතවේ.

Statement / ප්‍රකාශය C
This statement is correct. As it's built to unique specifications, custom software development is a time-consuming process that requires dedicated resources (developers, project managers, testers, etc.).
මෙම ප්‍රකාශය නිවැරදියි. එය අද්විතීය පිරිවිතරයන්ට අනුව ගොඩනගා ඇති බැවින්, අභිරුචි මෘදුකාංග සංවර්ධනය යනු කැපවූ සම්පත් (සංවර්ධකයින්, ව්‍යාපෘති කළමනාකරුවන්, පරීක්ෂකයින්, ආදිය) අවශ්‍ය වන කාලය ගතවන ක්‍රියාවලියකි.

Therefore, the correct answer is option 2).
එබැවින් නිවැරදි පිළිතුර විකල්පය 2) වේ.

2. Calculate the value of: / අගය ගණනය කරන්න:

$101.1_2 + 3.75_{10} \times 2_{10} - 7.A_{16}$

1) <u>5.375</u> 2) 6.625 3) 8.125 4) 10.875 5) 5.625

Answer : 1)

Explanation / පැහැදිලි කිරීම:

Step 1: Convert all numbers to decimal / සියලුම සංඛ්‍යා දශම සංඛ්‍යා බවට පරිවර්තනය කරන්න.

Since the answers are given in decimal format and calculations are simpler in decimal, we first convert all numbers from their respective bases into decimal.
පිළිතුරු දශම සංඛ්‍යා ආකෘතියෙන් ලබා දී ඇති නිසාත්, ගණනය කිරීම් දශම සංඛ්‍යා ආකෘතියෙන් සරල නිසාත්, පළමුව සියලුම සංඛ්‍යා ඒවායේ අදාළ පාදවලින් දශම සංඛ්‍යා බවට පරිවර්තනය කරමු.

101.1_2			
1	0	1	.1
2^2	2^1	2^0	$.2^{-1}$
1 × 4	0 × 2	1 × 1	1 × 0.5
4 + 0 + 1 + 0.5			
5.5_{10}			

$7.A_{16}$	
7	A = 10
16^0	$.16^{-1}$
7 × 1	10 × 0.0625
7 + 0.625	
7.625_{10}	

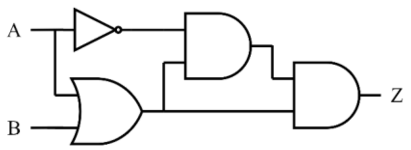
Step 2: Perform the Calculation / ගණනය කිරීම සිදු කරන්න.

After converting all values to decimal, we apply the BODMAS rule to solve the expression in the correct order.
සියලුම අගයන් දශමයට පරිවර්තනය කිරීමෙන් පසු, ප්‍රකාශනය නිවැරදි අනුපිළිවෙලට විසඳීමට BODMAS රීතිය යොදන්නෙමු.

$101.1_2 + 3.75_{10} \times 2_{10} - 7.A_{16}$
$= 5.5_{10} + 3.75_{10} \times 2_{10} - 7.625_{10}$
$= 5.5 + 7.50 - 7.625$
$= 13 - 7.625$
$= 5.375_{10}$

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

3. What is the Boolean expression that can be obtain from the below logic circuit?
පහත තාර්කික පරිපථයෙන් ලබාගත හැකි බූලීය ප්‍රකාශනය කුමක්ද?

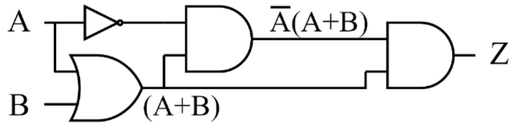


1) 1 2) 0 3) A 4) <u>$\bar{A}B$</u> 5) $(\bar{A}+B)(A.B)$

Answer: 4)

Explanation / පැහැදිලි කිරීම:

Let's obtained an expression from the circuit.
පරිපථයෙන් ප්‍රකාශනයක් ලබා ගනිමු.



Let's simplify the expression.
ප්‍රකාශනය සරල කර ගනිමු.

$Z=(A+B)(A+B)\bar{A}$

$=(AA+AB+AB+BB)\bar{A}$	Distributive law / විඝටන න්‍යාය
$=(A+AB+AB+BB)\bar{A}$	Idempotent law / තද්වභාවී න්‍යාය
$=(A+AB+BB)\bar{A}$ I	dempotent law / තද්වභාවී න්‍යාය
$=(A+AB+B)\bar{A}$	Idempotent law / තද්වභාවී න්‍යාය
$=(A(1+B)+B)\ \bar{A}$	Distributive law / විඝටන න්‍යාය
$=(A(1)+B)\bar{A}$	Identity law / සර්වසාම්‍ය න්‍යාය
$=(A+B)\bar{A}$	Identity law / සර්වසාම්‍ය න්‍යාය
$=A\bar{A}+\bar{A}B$	Distributive law / විඝටන න්‍යාය
$=0+\bar{A}B$	Inverse law / ප්‍රතිලෝම න්‍යාය
$=\bar{A}B$	Identity law / සර්වසාම්‍ය න්‍යාය

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

4. Which transition in operating system design most fundamentally enabled the move from monolithic execution to multi-user, multitasking environments, while also introducing process isolation?
මෙහෙයුම් පද්ධති නිර්මාණයේ කුමන සංක්‍රාන්තිය මොනොලිතික් ක්‍රියාත්මක කිරීමේ සිට බහු-පරිශීලක, බහුකාර්ය පරිසරයන් වෙත මාරුවීමට සහ ක්‍රියාවලි හුදකලාව හඳුන්වා දීමට මූලික වශයෙන් හැකියාව ලබා දුන්නේ දැයි

1) The introduction of batch processing systems in early mainframes
මුල් කාලීන ප්‍රධාන රාමු වල කණ්ඩායම් සැකසුම් පද්ධති හඳුන්වාදීම
2) <u>The development of time-sharing systems with memory protection</u>
මතක ආරක්ෂාව සහිත කාල විභජන පද්ධති සංවර්ධනය
3) The use of command-line interfaces in Unix-based systems
යුනික්ස් පාදක පද්ධතිවල විධාන රේඛා අතුරුමුහුණත් භාවිතය
4) The implementation of graphical user interfaces in personal computers
පුද්ගලික පරිගණකවල චිත්‍රක පරිශීලක අතුරුමුහුණත් ක්‍රියාත්මක කිරීම
5) The rise of mobile operating systems with app sandboxing
යෙදුම් sandboxing සමඟ ජංගම මෙහෙයුම් පද්ධතිවල නැගීම

Answer : 2)

Explanation / පැහැදිලි කිරීම:

Option / විකල්පය 1:
Incorrect. Batch processing introduced job scheduling but lacked interactive use and real-time multitasking. All tasks ran in sequence with no concept of memory protection or isolation.
වැරදියි. කණ්ඩායම් සැකසුම් මඟින් රැකියා කාලසටහන්ගත කිරීම හඳුන්වා දුන් නමුත් අන්තර්ක්‍රියාකාරී භාවිතය සහ තත්‍ය කාලීන බහුකාර්ය නොමැති විය. සියලුම කාර්යයන් මතක ආරක්ෂාව හෝ හුදකලාව පිළිබඳ සංකල්පයකින් තොරව අනුපිළිවෙලින් ක්‍රියාත්මක විය.

Option / විකල්පය 2:
Correct. Time-sharing systems allowed multiple users to run programs seemingly simultaneously. Combined with memory protection, these systems enforced process isolation, ensuring one process could not interfere with another. This was a foundational step toward modern multitasking operating systems.
නිවැරදි. කාල බෙදාගැනීමේ පද්ධති බහු පරිශීලකයින්ට එකවර වැඩසටහන් ක්‍රියාත්මක කිරීමට ඉඩ දුන්නේය. මතක ආරක්ෂාව සමඟ ඒකාබද්ධව, මෙම පද්ධති ක්‍රියාවලි හුදකලාව බලාත්මක කළ අතර, එක් ක්‍රියාවලියකට තවත් ක්‍රියාවලියකට බාධා කළ නොහැකි බව සහතික කළේය. මෙය නවීන බහුකාර්ය මෙහෙයුම් පද්ධති සඳහා මූලික පියවරක් විය.

Option / විකල්පය 3:
Incorrect. Command-line interfaces improved user interaction but did not structurally change how the OS managed processes or memory. It was an interface innovation, not a core architecture shift.
වැරදියි. විධාන රේඛා අතුරුමුහුණත් පරිශීලක අන්තර්ක්‍රියා වැඩිදියුණු කළ නමුත් මෙහෙයුම් පද්ධතිය ක්‍රියාවලි හෝ මතකය කළමනාකරණය කරන ආකාරය ව්‍යුහාත්මකව වෙනස් කළේ නැත. එය අතුරුමුහුණත් නවෝත්පාදනයක් වූ අතර මූලික ගෘහ නිර්මාණ මාරුවක් නොවේ.

Option / විකල්පය 4:
Incorrect. Graphical user interfaces improved usability but were built on top of systems that already supported multitasking. GUIs were a user-level improvement, not an architectural leap.
වැරදියි. චිත්‍රක පරිශීලක අතුරුමුහුණත් භාවිතය වැඩි දියුණු කළ නමුත් දැනටමත් බහුකාර්ය සඳහා සහය දැක්වූ පද්ධති මත ගොඩනගා ඇත. GUI යනු වාස්තු විද්‍යාත්මක පිම්මක් නොව පරිශීලක මට්ටමේ වැඩිදියුණු කිරීමකි.

Option / විකල්පය 5:
Incorrect. App sandboxing is a modern extension of process isolation principles. While important, it builds upon the core architectural concepts introduced in earlier time-sharing systems.
වැරදියි. යෙදුම් වැලිපිල්ල යනු ක්‍රියාවලි හුදකලා මූලධර්මවල නවීන දිගුවකි. වැදගත් වුවද, එය පෙර කාල බෙදාගැනීමේ පද්ධතිවල හඳුන්වා දුන් මූලික ගෘහ නිර්මාණ සංකල්ප මත ගොඩනගා ඇත.

Therefore, the correct answer is 2).
එබැවින්, නිවැරදි පිළිතුර 2) වේ.

5. What does a **CHECK** constraint ensure in a table column?
වගු තීරුවක **CHECK** සංරෝධකය සහතික කරන්නේ කුමක්ද?

```
CREATE TABLE employees (
              emp_id INT PRIMARY KEY,
              salary DECIMAL(10, 2) CHECK (salary >= 3000));
```

1) Ignores salary values.
2) <u>Enforces `salary >= 3000`.</u>
3) Sets salary to 3000.
4) Allows any salary.
5) Deletes rows with low salary.

Answer : 2)

A CHECK constraint in a SQL table column ensures that the values inserted into that column meet a specific condition. In the given example, the CHECK constraint CHECK (`salary >= 3000`) ensures that the salary column can only contain values that are greater than or equal to 3000. If a user tries to insert a value into the salary column that is less than `3000`, the database will raise an error, preventing the insertion.
SQL වගු තීරුවක ඇති CHECK සම්බාධකය මඟින් එම තීරුවට ඇතුළු කර ඇති අගයන් නිශ්චිත කොන්දේසියක් සපුරාලන බව සහතික කරයි. ලබා දී ඇති උදාහරණයේ, CHECK සම්බාධකය CHECK (`salary >= 3000`) වැටුප් තීරුවේ අඩංගු විය හැක්කේ 3000ට වඩා වැඩි හෝ සමාන අගයන් පමණක් බව සහතික කරයි. පරිශීලකයෙකු 3000ට අඩු වැටුප් තීරුවට අගයක් ඇතුළු කිරීමට උත්සාහ කරන්නේ නම්, දත්ත සමුදාය දෝෂයක් මතු කරයි, ඇතුළත් කිරීම වළක්වයි.

Therefore, the correct answer is 2).
එබැවින්, නිවැරදි පිළිතුර 1) වේ.

6. What will be the result of the following Python code?
පහත පයිතන් කේතයේ ප්‍රතිඵලය කුමක් වේද?

```
def mystery(x, y):
    x = x + y
    y = y + x
    return x, y
x, y = 10, 20
result = mystery(x, y)
print(x, y, result)
```

1) `10 20 (30, 50)` 2) `30 20 (30, 50)` 3) `10 20 (30, 30)`
4) `30 50 (30, 50)` 5) `10 20 (20, 40)`

Answer : 1)

This Python code defines a function `mystery` that takes two parameters `x` and `y`. Inside the function, it first updates `x` to be the sum of the original `x` and `y`, and then updates `y` to be the sum of the updated `y` and the new value of `x`.
මෙම පයිතන් කේතය x සහ y පරාමිති දෙකක් ගන්නා ශ්‍රිතය mystery නිර්වචනය කරයි. ශ්‍රිතය තුළ, එය මුලින්ම x යනු මුල් x සහ y හි එකතුව ලෙස යාවත්කාලීන කරයි, පසුව y යාවත්කාලීන කරන ලද y හි එකතුව සහ x හි නව අගය ලෙස යාවත්කාලීන කරයි.

After executing the function, it returns the new values of `x` and `y`. When the function is called with `x = 10` and `y = 20`, inside the function, x becomes `10 + 20 = 30`, and y becomes `20 + 30 = 50`.
ශ්‍රිතය ක්‍රියාත්මක කිරීමෙන් පසුව, එය x සහ y හි නව අගයන් ලබා දෙයි. ශ්‍රිතය x = 10 සහ y = 20 සමඟ හැඳින්වූ විට, ශ්‍රිතය තුළ, x 10 + 20 = 30 බවටත්, y 20 + 30 = 50 බවටත් පත් වේ.

However, these changes are `local` to the function and do not affect the variables outside it. Thus, when printing `x` and `y` after the function call, they remain as their original values, `10` and `20`, respectively.
කෙසේ වෙතත්, මෙම වෙනස්කම් ශ්‍රිතයට local වන අතර ඉන් පිටත විචල්‍යයන්ට බලපාන්නේ නැත. මේ අනුව, ශ්‍රිත ඇමතුමෙන් පසුව x සහ y මුද්‍රණය කරන විට, ඒවා පිළිවෙලින් 10 සහ 20 ලෙස ඒවායේ මුල් අගයන් ලෙස පවතී.

The result variable captures the returned tuple `(30, 50)`. Therefore, the output of the code will be `10 20 (30, 50)`.
ප්‍රතිඵල විචල්‍යය මඟින් ආපසු ලබා දුන් එමවකෑ (30, 50) ග්‍රහණය කරයි. එබැවින්, කේතයේ ප්‍රතිදානය 10 20 (30, 50) වනු ඇත.

Therefore, the correct answer is 1) `10 20 (30, 50)`
එබැවින්, නිවැරදි පිළිතුර 1) `10 20 (30, 50)` වේ.

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

```
for i in range(2):
    count = 1
    while count <= 2:
          print(i, count)
          count += 1
```

1)	2)	3)	4)	5)
0 1	0 1	0 1	0 1	1 1
0 2	1 1	0 2	1 1	1 2
1 1	0 2	1 1	2 1	2 1
1 2	1 2			2 2

Answer : 1)

`for i in range(2):`
This line starts a loop where the variable `i` will take values from `0` up to, but not including, 2. So, `i` will be `0` in the first loop and `1` in the second loop.
මෙම පේළිය ලූපයක් ආරම්භ කරන අතර එහිදී `i` විචල්‍යය `0` සිට අගයන් ගනී, නමුත් 2 ඇතුළත් නොකරයි. එබැවින්, පළමු ලූපයේ `i 0` වන අතර දෙවන ලූපයේ `1` වනු ඇත.

`count = 1`
Inside the `for` loop, `count` is set to `1` at the beginning of each iteration. This means every time `i` changes, `count` resets to `1`.
`for` ලූපය තුළ, සෑම පුනරාවර්තනයකම ආරම්භයේ දී `count 1` ලෙස සකසා ඇත. මෙයින් අදහස් කරන්නේ `i` වෙනස් වන සෑම අවස්ථාවකම, `count 1` ලෙස නැවත සකසන බවයි.

`while count <= 2:`
This line starts a `while` loop that will keep running as long as `count` is less than or equal to `2`. So for each `i`, the loop will run twice because count starts at `1` and increases.
මෙම පේළිය `while` ලූපයක් ආරම්භ කරන අතර එය `count 2` ට වඩා අඩු හෝ සමාන වන තාක් කල් ක්‍රියාත්මක වේ. එබැවින් සෑම `i` සඳහාම, loop එක දෙවරක් ධාවනය වනු ඇත, මන්ද `count 1` න් ආරම්භ වී වැඩි වේ.

`print(i, count)`
Inside the `while` loop, this line prints the current values of `i` and `count` together. So it shows which iteration of the `for` and `while` loops we are in.
`while` ලූපය තුළ, මෙම පේළිය `i` සහ `count` හි වත්මන් අගයන් එකට මුද්‍රණය කරයි. එබැවින් එය අප සිටින්නේ `for` සහ `while` ලූපවල කුමන පුනරාවර්තනයකද යන්න පෙන්වයි.

`count += 1`
This line increases the value of `count` by `1` after printing. It moves the `while` loop closer to its ending condition by incrementing count each time.
මෙම පේළිය මුද්‍රණය කිරීමෙන් පසු `count` හි අගය `1` කින් වැඩි කරයි. එය සෑම අවස්ථාවකම `count` වැඩි කිරීමෙන් `while` ලූපය එහි අවසන් තත්ත්වයට සමීප කරයි.

Therefore, the answer is 1)
එබැවින් පිළිතුර 1)

8. What system development model is appropriate for systems where the requirement is clear and consistent?
අවශ්‍යතාවය පැහැදිලි සහ ස්ථාවර වන පද්ධති සඳහා සුදුසු වන පද්ධති සංවර්ධන ආකෘතිය කුමක් ද?

1) Waterfall model / දියඇලි ආකෘතිය
2) Spiral model / සර්පිල ආකෘතිය
3) Agile model / සුචලය ආකෘතිය
4) Prototyping / මූලාකෘතිකරණය
5) Rapid Application development / ශීඝ්‍ර යෙදුම් සංවර්ධනය

Answer: 1)

Let's consider each option one by one. / එක් එක් විකල්පය එකින් එක සලකා බලමු.

Statement / ප්‍රකාශය 1)
According to this statement, it's true. The Waterfall model is appropriate for systems where the requirements are clear and consistent. It is a linear and sequential approach where each phase of development (requirements, design, implementation, testing, deployment, and maintenance) is completed before moving on to the next. This model works well when the project requirements are well-understood and unlikely to change.
මෙම ප්‍රකාශය අනුව, එය සත්‍ය වේ. දියඇලි ආකෘතිය අවශ්‍යතා පැහැදිලි සහ ස්ථාවර පද්ධති සඳහා සුදුසු වේ. එය රේඛීය සහ අනුක්‍රමික ප්‍රවේශයක් වන අතර එහිදී සංවර්ධනයේ එක් එක් අදියර (අවශ්‍යතා, සැලසුම් කිරීම,ක්‍රියාත්මක කිරීම,පරීක්ෂා කිරීම,යෙදවීම සහ නඩත්තු කිරීම) ඊළඟට යාමට පෙර අවසන් වේ. ව්‍යාපෘති අවශ්‍යතා හොඳින් තේරුම් ගෙන වෙනස් වීමට ඉඩ ඇති විට මෙම ආකෘතිය හොඳින් ක්‍රියා කරයි.

Statement / ප්‍රකාශය 2)
According to this statement, it's false. The Spiral model is more suitable for projects where requirements are not well-defined or are expected to evolve. It combines iterative development with risk analysis is ideal for projects with high uncertainty or where requirements may change over time.
මෙම ප්‍රකාශයට අනුව එය අසත්‍ය වේ. අවශ්‍යතා හොඳින් අර්ථ දක්වා නොමැති හෝ පරිණාමය වීමට අපේක්ෂා කරන ව්‍යාපෘති සඳහා සර්පිලාකාර ආකෘතිය වඩාත් සුදුසු වේ. එය අවදානම් විශ්ලේෂණය සමඟ පුනරාවර්තන සංවර්ධනය ඒකාබද්ධ කරන අතර ඉහළ අවිනිශ්චිතතාවයක් ඇති හෝ කාලයත් සමඟ අවශ්‍යතා වෙනස් විය හැකි ව්‍යාපෘති සඳහා වඩාත් සුදුසු වේ.

Statement / ප්‍රකාශය 3)
According to this statement, it's false. The Agile model is designed for projects where requirements are expected to evolve or are not fully understood from the start. It emphasizes iterative development, flexibility, and customer collaboration, making it less appropriate for projects with clear and consistent requirements.
මෙම ප්‍රකාශයට අනුව එය අසත්‍ය වේ. Agile ආකෘතිය නිර්මාණය කර ඇත්තේ අවශ්‍යතා පරිණාමය වීමට අපේක්ෂා කරන හෝ ආරම්භයේ සිට සම්පූර්ණයෙන් වටහා නොගත් ව්‍යාපෘති සඳහා ය. එය පුනරාවර්තන සංවර්ධනය, නම්‍යශීලී බව සහ පාරිභෝගික සහයෝගිතාවය අවධාරණය කරයි, පැහැදිලි සහ ස්ථාවර අවශ්‍යතා සහිත ව්‍යාපෘති සඳහා එය අඩු යෝග්‍ය බවට පත් කරයි.

Statement / ප්‍රකාශය 4)
According to this statement, it's false. Prototyping involves creating early versions of a system to explore ideas or refine requirements. It is particularly useful when requirements are unclear or need refinement. For projects with clear and consistent requirements, prototyping may not be necessary.
මෙම ප්‍රකාශයට අනුව එය අසත්‍ය වේ. මූලාකෘතිකරණය යනු අදහස් ගවේෂණය කිරීමට හෝ අවශ්‍යතා පිරිපහදු කිරීමට පද්ධතියක මුල් අනුවාද නිර්මාණය කිරීමයි. අවශ්‍යතා අපැහැදිලි හෝ ශෝධනය අවශ්‍ය වූ විට එය විශේෂයෙන් ප්‍රයෝජනවත් වේ. පැහැදිලි සහ ස්ථාවර අවශ්‍යතා සහිත ව්‍යාපෘති සඳහා, මූලාකෘති කිරීම අවශ්‍ය නොවනු ඇත.

Statement / ප්‍රකාශය 5)
According to this statement, it's false. RAD focuses on rapid development and iteration, often using prototypes and user feedback to quickly refine the system. It is more suited to projects where requirements may change or are not fully defined at the outset.
මෙම ප්‍රකාශයට අනුව එය අසත්‍ය වේ. RAD වේගවත් සංවර්ධනය සහ පුනරාවර්තනය කෙරෙහි අවධානය යොමු කරයි, පද්ධතිය ඉක්මනින් පිරිපහදු කිරීම සඳහා බොහෝ විට මූලාකෘති සහ පරිශීලක ප්‍රතිපෝෂණ භාවිතා කරයි. අවශ්‍යතා වෙනස් විය හැකි හෝ ආරම්භයේදී සම්පූර්ණයෙන් අර්ථ දක්වා නොමැති ව්‍යාපෘති සඳහා එය වඩාත් සුදුසු වේ.

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

9. Which of the following PHP statements correctly creates a MySQL connection in object-oriented style?
පහත සඳහන් PHP ප්‍රකාශනවලින් කුමන ප්‍රකාශය වස්තු-නැඹුරු ශෛලියෙන් MySQL සම්බන්ධතාවයක් නිවැරදිව නිර්මාණය කරයිද?

1) `$conn = mysqli("localhost", "root", "");`
2) `$conn = new mysqli("localhost", "root", "");`
3) `$conn = connect_mysqli("localhost", "root", "");`
4) `$conn = mysqli_connect("localhost", "root", "");`
5) `$conn = db_connect("localhost", "root", "");`

Answer: 2)

Explanation / පැහැදිලි කිරීම:

In PHP, there are two main styles to connect to MySQL:
PHP හි, MySQL වෙත සම්බන්ධ වීමට ප්‍රධාන විලාස දෙකක් තිබේ:

Procedural style → uses functions like `mysqli_connect()`.
ක්‍රියා පටිපාටි විලාසය → `mysqli_connect()` වැනි ශ්‍රිත භාවිතා කරයි.

Object-oriented style → uses the `mysqli` class with `new`.
වස්තු-නැඹුරු ශෛලිය → `new` සමඟ `mysqli` පන්තිය භාවිතා කරයි.

Option / විකල්ප 1)
Wrong. `mysqli` is a class, not a function, so you can’t call it like this.
වැරදියි. `mysqli` යනු පන්තියක් මිස ශ්‍රිතයක් නොවේ, එබැවින් ඔබට එය මේ ආකාරයෙන් හැඳින්විය නොහැක.

Option / විකල්ප 2)
Correct. This is the object-oriented style:
නිවැරදියි. මෙය වස්තු-නැඹුරු විලාසයයිට

`"localhost"` → `host`, `"root"` → `username`, `""` → `password` (empty in this case / මෙම අවස්ථාවේදී හිස්), `"database_name"` (optional, can be added / එකතු කළ හැක).

Option / විකල්ප 3)
Wrong. There’s no such function `connect_mysqli`.
වැරදියි. `connect_mysqli` වැනි ශ්‍රිතයක් නොමැත.

Option / විකල්ප 4)
This is valid, but it’s procedural style, not object-oriented.
මෙය වලංගු වේ, නමුත් එය ක්‍රියා පටිපාටි විලාසයයි, වස්තු-නැඹුරු නොවේ.

Option / විකල්ප 5)
Wrong. There’s no built-in function `db_connect` in PHP. වැරදියි. PHP හි `db_connect` ශ්‍රිතයක් නොමැත.

Therefore, the answer is 2)
එබැවින්, පිළිතුර 2) වේ.

10. Which CSS property is used to control the spacing between the lines of text within an element?
මූලාංගයක් තුළ ඇති පාඨ රේඛා අතර පරතරය පාලනය කිරීමට භාවිතා කරන CSS ගුණාංගය කුමක් ද?

1) `letter-spacing` 2) `line-height` 3) `text-indent` 4) `word-spacing` 5) `margin`

Answer : 2)

The CSS property used to control the spacing between lines of text within an element is called `line-height`.
මූලාංගයක් තුළ පාඨ රේඛා අතර පරතරය පාලනය කිරීමට භාවිතා කරන CSS උපලක්ෂණය `line-height` ලෙස හැඳින්වේ.

Here's how `line-height` works / මෙන්න `line-height` ක්‍රියා කරන ආකාරය:

It specifies the height of each line box within an element. It can be set to a specific unit value (like pixels or percentages) or a unitless number.
එය මූලාංගයක් තුළ එක් එක් රේඛා පෙට්ටියේ උස නියම කරයි. එය නිශ්චිත ඒකක අගයකට (පික්සල හෝ ප්‍රතිශත වැනි) හෝ ඒකක රහිත සංඛ්‍යාවකට සැකසිය හැක.

A unitless number (ex: 1.5) is multiplied by the font size of the element to determine the line height.
ඒකක රහිත අංකයක් (උදා: 1.5) රේඛාවේ උස තීරණය කිරීම සඳහා මූලාංගයේ අකුරු ප්‍රමාණයෙන් ගුණ කරනු ලැබේ.

`line-height` affects the vertical spacing between lines of text, making the text more readable and improving the overall layout.
`line-height` පෙළ රේඛා අතර සිරස් පරතරයට බලපාන අතර, පෙළ වඩාත් කියවිය හැකි කර සමස්ත පිරිසැලසුම වැඩිදියුණු කරයි.

This property is particularly useful for adjusting the spacing between lines of text to improve readability and aesthetics on a webpage.
වෙබ් පිටුවක කියවීමේ හැකියාව සහ සෞන්දර්යය වැඩිදියුණු කිරීම සඳහා පාඨ රේඛා අතර පරතරය සකස් කිරීම සඳහා මෙම ගුණාංගය විශේෂයෙන් ප්‍රයෝජනවත් වේ.

`letter-spacing` is a CSS property that controls the amount of space between characters in text. It allows you to increase or decrease the space between letters, making text tighter or more spread out. Therefore, answer 1) is wrong.
`letter-spacing` යනු පෙළෙහි අක්ෂර අතර ඉඩ ප්‍රමාණය පාලනය කරන CSS ගුණාංගයකි. එය ඔබට අකුරු අතර අවකාශය වැඩි කිරීමට හෝ අඩු කිරීමට ඉඩ සලසයි, පෙළ තද කිරීම හෝ පැතිරීම සිදු කරයි. එබැවින් පිළිතුර 1 වැරදියි.

`text-indent` is a CSS property that specifies the indentation of the first line of text within an element. It allows you to push the first line of a block of text inward, creating an indentation effect. Therefore answer 3) is wrong.
`text-indent` යනු මූලාංගයයක් තුළ ඇති පෙළෙහි පළමු පේළියේ එබුම සඳහන් කරන CSS ගුණාංගයකි. ඉන්ඩෙන්ටේෂන් ආවරණයක් නිර්මාණය කරමින් පෙළ කොටසක පළමු පේළිය ඇතුළට තල්ලු කිරීමට එය ඔබට ඉඩ සලසයි. එබැවින් පිළිතුර 3 වැරදියි

`word-spacing` is a CSS property that controls the spacing between words in a block of text. It allows you to increase or decrease the amount of space between words to adjust the overall appearance of text content. Therefore answer 4) is wrong.
`word-spacing` යනු පෙළ කොටසක වචන අතර පරතරය පාලනය කරන CSS ගුණයකි. පාඨ අන්තර්ගතයේ සමස්ත පෙනුම සකස් කිරීම සඳහා වචන අතර ඉඩ ප්‍රමාණය වැඩි කිරීමට හෝ අඩු කිරීමට එය ඔබට ඉඩ සලසයි. එබැවින් පිළිතුර 4 වැරදියි.

`margin` is a property that controls the space outside the border of an element. It specifies the amount of empty space that should be left around the outside of the element's content, padding, and border. Therefore answer 5) is wrong.
`margin` යනු මූලාංගයක මායිමෙන් පිටත අවකාශය පාලනය කරන ගුණාංගයකි. එය මූලාංගයේ අන්තර්ගතය, පිරවුම සහ මායිමේ පිටත ඉතිරි කළ යුතු හිස් ඉඩ ප්‍රමාණය නියම කරයි. එබැවින් පිළිතුර 5 වැරදියි.

Therefore, the correct answer is 2). / ඔබැවින්, නිවැරදි පිළිතුර 2) වේ.