Open quiz in app:
https://app.hoppy.day/q/abbwtu

Programming Fundamentals and Python Applications

Name: _____

Class: _____

1. Consider the following Python code segment:

```python
X = 23
Y = 5
Z = X // Y + X % Y
print(Z)
```

What is the value of Z?

A. 4

B. 3

C. 7

D. 4.6

2. In HKDSE ICT, which of the following statements about constants is CORRECT?

A. A constant's value can be updated using an assignment statement at any time.

B. A constant must only store numeric values.

C. A constant's value remains unchanged throughout the program execution.

D. Using constants makes a program more difficult to debug than using variables.

3. Consider the following list in Python:

```python
marks = [45, 62, 88, 31, 75]
```

A student wants to find the average of the marks in the list. Which of the following code segments is CORRECT? Assume $N = 5$.

A.

```python
total = 0
for i in range(0, N):
    total = total + marks[i]
avg = total / N
```

B.

```python
total = 0
for i in range(0, N):
    total = total + marks[i]
    avg = total / N
```

C.

```python
total = 0
for i in range(0, N):
    total = total + i
avg = total / N
```

D.

```python
total = marks[0] + marks[N-1]
avg = total / N
```

4. Evaluate the following HKDSE pseudocode segment:

Pseudocode

```
A ← 8
B ← 12
C ← 15
if (A + B > C) AND (C <> 15) then
   output "Alpha"
else
   if (B < C) OR (A > B) then
      output "Beta"
   else
      output "Gamma"
```

What is the output?

A. Alpha

B. Beta

C. Gamma

D. Nothing is output

5. What is the output of the following Python program segment?

Python

```
count = 0
for i in range(1, 10, 3):
   if i % 2 == 0:
      count = count + i
   else:
      count = count + 1
print(count)
```

A. 4

B. 5

C. 6

D. 12

6. Assume a Python list

```python
items
```

contains N elements. A student writes the following code to find the maximum value in the list:

```python
# Line 1: m = items[0]
# Line 2: for i in range(1, N):
# Line 3:     if items[i] < m:
# Line 4:         m = items[i]
```

The code incorrectly finds the minimum instead of the maximum. Which line needs to be changed to fix the logic?

A. Line 1

B. Line 2

C. Line 3

D. Line 4

7. Consider the following list transformation in Python:

```python
S = [2, 4, 6, 8, 10]
for i in range(0, 5):
    if S[i] > 5:
        S[i] = S[i] * 2
    else:
        S[i] = S[i] + 1
```

What is the final state of list S?

A. [4, 8, 12, 16, 20]

B. [3, 5, 12, 16, 20]

C. [3, 5, 6, 8, 10]

D. [2, 4, 12, 16, 20]

答案

1 C

This question tests the use of arithmetic operators in Python.

Step 1: Floor Division (//)
The

```python
//
```

operator performs integer division and discards the remainder.
$23//5 = 4$.

Step 2: Modulus (%)
The

```python
%
```

operator returns the remainder of the division.
$23 \pmod 5 = 3$ (since $5 \times 4 = 20$, $23 - 20 = 3$).

Step 3: Addition
$4 + 3 = 7$.

Option Analysis:
A: 4 is just the quotient.
B: 3 is just the remainder.
C: Correct. $4 + 3 = 7$.
D: 4.6 would be the result of a single float division $23/5$.

2 C

Let's review the fundamental properties of constants in programming.

Key Rule: A constant is an identifier whose value is assigned once and CANNOT be changed during the execution of the program. This helps prevent accidental bugs and improves readability.

Why other options are wrong:
A: This is the definition of a variable, not a constant.
B: While some languages have naming conventions for constants (like all caps), it is not a technical requirement for them to be numeric; they can be strings too.
D: Constants are useful precisely because they prevent values from being overwritten, improving code safety.

3 A

To calculate an average, you must sum all elements and then divide by the count.

Logic Check:
1. Initialize a sum variable (e.g.,

```python
total
```

) to 0.
2. Iterate through every element in the list using a loop.
3. Add each element

```python
marks[i]
```

to the total.
4. Divide the final total by the number of elements N.

Option Analysis:
A: Correct. It perfectly follows the summing and dividing logic.
B: Incorrect. It calculates the average inside the loop, which is inefficient and logically incorrect for the final result.
C: Incorrect. It adds the index i instead of the value in the list

```python
marks[i]
```

.
D: Incorrect. It only adds the first and last elements, ignoring the rest of the list.

4 B

Let's trace the logic with $A = 8, B = 12, C = 15$.

Step 1: First IF condition

```pseudocode
(A + B > C) AND (C <> 15)
```

- $8 + 12 > 15$ is $20 > 15$, which is TRUE.
- $15 <> 15$ (where

```pseudocode
<>
```

means not equal to) is FALSE.
-

```pseudocode
TRUE AND FALSE
```

results in FALSE.

Step 2: ELSE block (Outer)
Now we check the second IF condition:

```pseudocode
(B < C) OR (A > B)
```

- $12 < 15$ is TRUE.
- $8 > 12$ is FALSE.
-

```pseudocode
TRUE OR FALSE
```

results in TRUE.

Step 3: Conclusion
Since the second condition is True, the program outputs "Beta".

5 C

The

```python
range(start, stop, step)
```

function starts at 1, increments by 3, and stops before 10.

Iteration Trace:
1. $i = 1$:
- $1 \pmod 2 == 0$ is False.
-

```python
count = 0 + 1 = 1
```

.
2. $i = 4$:
- $4 \pmod 2 == 0$ is True.
-

```python
count = 1 + 4 = 5
```

.
3. $i = 7$:
- $7 \pmod 2 == 0$ is False.
-

```python
count = 5 + 1 = 6
```

.
4. The next value would be 10, which is not < 10, so the loop ends.

Final

```python
count
```

is 6.

6 C

To find the maximum, you must check if the current item is greater than our stored maximum m.

Analysis:
- Line 1 initializes m to the first element (Correct).
- Line 2 loops through the rest of the list (Correct).
- Line 3 currently uses

```python
<
```

, which means it updates m whenever it finds a smaller number (this leads to finding the minimum).
- To find the maximum, the condition should be

```python
if items[i] > m:
```

.

Therefore, Line 3 contains the logic error.

7 B

Let's iterate through each element of the list $S = [2, 4, 6, 8, 10]$:

1. $i = 0, S[0] = 2$: $2 > 5$ is False. $S[0] = 2 + 1 = 3$.
2. $i = 1, S[1] = 4$: $4 > 5$ is False. $S[1] = 4 + 1 = 5$.
3. $i = 2, S[2] = 6$: $6 > 5$ is True. $S[2] = 6 \times 2 = 12$.
4. $i = 3, S[3] = 8$: $8 > 5$ is True. $S[3] = 8 \times 2 = 16$.
5. $i = 4, S[4] = 10$: $10 > 5$ is True. $S[4] = 10 \times 2 = 20$.

Result:

```python
[3, 5, 12, 16, 20]
```

.