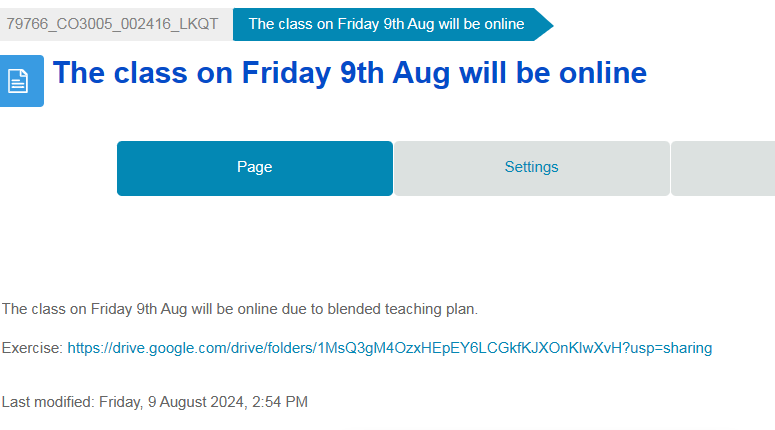
**Please go to LMS system to get questions**

**I will be back at 15:15**



**Question 1.** Which of the following Java bytecode instructions is used to return an integer from a method?

**- A) `ireturn`**

- B) `areturn`

- C) `dreturn`

- D) `freturn`

**Question 2.**Given a method declared in class Y, written in Java, as follows:

`public static void bar(double d, B b, int[] arr);`

Where `B` is a previously declared class. The minimum size of the local variable array needed is:

- A) 3

**- B) 4**

- C) 5

- D) 6

Double: 2 bytes

Int: 1 byte, consider only allocate

B: at least 1 byte for Boolean

Because the JVM uses a 32-bit architecture for its stack frames and local variable arrays, any data type that is larger than 32 bits occupies more than one "slot" in the local variable array.

double is 64 bits so it takes 2 slot

**Question 3.** Consider the following Java code snippet:

```java

boolean result = a || (b && c);

```

The corresponding bytecode would be:

1. `iload\_1`a

2. `ifne Label1` if not equal 0

3. `iload\_2` b

4. `ifne Label1`

5. `iload\_3`c

6. `ifne Label2` and

7. `Label1:`

8. \_\_\_\_\_\_\_

9. `Label2:`

…

The code at line 8 should be:

- A) `**iconst\_1**`

- B) `iconst\_0`

- C) `istore\_1`

- D) `istore\_2`

**Question 4.**In a given function:

```java

void foo(int a, int b) {

a = b + 1;

b = a - 2;

}

```

If `a` and `b` are stored in local variable indices 0 and 1 respectively, and assuming short-circuit evaluation, what is the correct bytecode instruction to store the result of `b = a - 2`?

- A) `istore\_0`

**- B) `istore\_1` store b**

- C) `iload\_2`

- D) `iload\_1`

**Question 5.** The effect of short-circuiting is most evident at line \_\_\_ of the provided bytecode sequence.

- A) 2

**- B) 4**

- C) 6

- D) 8

**Question 6.** Which of the following statements about lazy evaluation is true?

- A) It avoids unnecessary calculations

**- B) It is always less efficient than eager evaluation**

- C) It is mostly used to maximize CPU usage

- D) It is not used in functional programming models

Use the following code to answer questions 7 to 10:

```python

def example\_function(a, b, c):

a = a + b

b = b + c

c = a + c

print(a, b, c)

x = 2

y = 3

z = 4

example\_function(x, y, z)

print(x, y, z)

```

**Question 7.** If all parameters are passed by value, the printed values are:

- A) `5 7 9` and `2 3 4`

- B) `5 7 9` and `5 7 9`

- C) `2 3 4` and `5 7 9`

- D) `2 3 4` and `2 3 4`

Passed by value

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | x | y | z | a | b | c |
|  | 2 | 3 | 4 |  |  |  |
|  | 2 | 3 | 4 | 2 | 3 | 4 |
|  |  |  |  | 5 |  |  |
|  |  |  |  |  | 7 |  |
|  |  |  |  |  |  | 9 |
|  | 2 | 3 | 4 | 5 | 7 | 9 |

=> example\_function(x, y, z) = ‘5 7 9’

print(x, y, z) = ‘2 3 4’

**Question 8.** If parameters `a` and `b` are passed by value-result and `c` is passed by reference, the printed values are:

**- A) `5 7 9` and `5 7 9`**

- B) `5 7 11` and `5 7 11`

- C) `5 7 11` and `2 3 11`

- D) `2 3 9` and `2 3 4`

Passed by reference modifies the parameter input.

Passed by value-result allows return the current parameter to corresponding parameter input

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | x | y | z | a | b | c |
|  | 2 | 3 | 4 |  |  |  |
|  | 2 | 3 | 4 | 2 | 3 | 4 |
|  |  |  |  | 5 |  |  |
|  |  |  |  |  | 7 |  |
|  |  |  | **9** |  |  | 9 |
|  | **5** | **7** | 9 | 5 | 7 | 9 |

=> example\_function(x, y, z) = ‘5 7 9’

print(x, y, z) = ‘5 7 9’

**Question 9.** If `a` is passed by reference, `b` by value, and `c` by pointer, the printed values are:

**- A) `5 7 7` and `5 3 7`**

- B) `5 7 11` and `5 3 11`

- C) `7 7 7` and `2 3 7`

- D) `7 7 11` and `2 3 11`

Pass by pointer: Maybe

b = b + c, since c is pointer, c overshadows b and write 7 to b

Then c = c + a, since c is pointer, c overshadows a and write 7 to c

Then in function it prints '5 7 7' but since b is passed by value, outside function it prints '5 3 7'

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | x | y | z | a | b | c |
|  | 2 | 3 | 4 |  |  |  |
|  | 2 | 3 | 4 | 2 | 3 | 4 |
|  | **5** |  |  | 5 |  |  |
|  |  |  |  |  | **7** |  |
|  |  |  | **7** |  |  | **7** |
|  | 5 | 3 | 7 | 5 | 7 | 7 |

=> example\_function(x, y, z) = ‘5 7 7’

print(x, y, z) = ‘5 3 7’

**Question 10.** If all parameters are passed by name, the printed values are:

- A) `5 7 11` and `5 7 11`

**- B) `5 7 9` and `2 3 4`**

- C) `7 10 14` and `7 10 14`

- D) `7 10 14` and `2 3 4`

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | x | y | z | a | b | c |
|  | 2 | 3 | 4 |  |  |  |
|  | 2 | 3 | 4 | 2 | 3 | 4 |
|  |  |  |  | 5 |  |  |
|  |  |  |  |  | 7 |  |
|  |  |  | **9** |  |  | 9 |
|  | **5** | **7** | 9 | 5 | 7 | 9 |

=> example\_function(x, y, z) = ‘5 7 9’

print(x, y, z) = ‘2 3 4’