(/)

Curriculum
SE Foundations ⌃
Average: **108.76%** ⌄

Evaluation quiz correction

Evaluation Quiz: Evaluation #4

Date: 2023-07-03

Status: Done

Duration: 24 minutes

Score: 66.67%

 # "I don't know": 0

 # Success: 12

 # Fail: 6

Responses

0. In a singly linked list, what are possible directions to traverse it?

(select all possible answers)

Score: 0.0

- ☑ **Forward**
- ☑ **Backward**
- ☐ I don't know

1. Is this module correctly documented?

Help

```
#!/usr/bin/python3
"""()
    My calculation module
"""
import sys
...
```

Score: 0.0

☐ **Yes**

☑ **No**

☐ I don't know

2. Given this code:

```
struct point {
    int x;
    int y;
};
struct point my_point = { 3, 7 };
struct point *p = &my_point;
```

To set the member y of my variable my_point to 98, I can do (select all valid answers):

Score: 0.0

☐ **my_point.y = 98**

☐ my_point->y = 98

☐ p.y = 98

☑ **(*p).y = 98**

☑ **p->y = 98**

☐ I don't know

3. Is this a standardized way to comment a function in Python?

```
/* Addition function */
def add(a, b):
    return a + b
```

Score: 1.0

☑ **No**
  (/)
☐ Yes

☐ I don't know

4. Bubble Sort is a _____.

Score: 1.0

☑ **simple comparison sorting algorithm**

☐ complex comparison sorting algorithm

☐ simple non-comparison searching algorithm

☐ simple non-comparison sorting algorithm

☐ I don't know

5. In this following code, what is `__password` ?

```
class User:
    id = 89
    name = "no name"
    __password = None

    def __init__(self, new_name=None):
        self.is_new = True
        if new_name is not None:
            self.name = new_name
```

Score: 0.0

☐ A public class attribute

☐ A public instance attribute

☐ A protected class attribute

☐ A protected instance attribute

☐ **A private class attribute**

☑ **A private instance attribute**

☐ I don't know

6. What is `__repr__` ? (/)

Score: 1.0

☐ Instance method that prints an "official" string representation of an instance

☑ **Instance method that returns an "official" string representation of an instance**

☐ Instance method that returns the dictionary representation of an instance

☐ I don't know

7. What do these lines print?

```python
class Base():
    """ My base class """

    __nb_instances = 0

    def __init__(self):
        Base.__nb_instances += 1
        self.id = Base.__nb_instances

class User(Base):
    """ My User class """

    def __init__(self):
        super().__init__()
        self.id += 99

u = User()
print(u.id)
```

Score: 1.0

☐ 99

☑ **100**

☐ 1

☐ I don't know

8. What is `__doc__` ?

Score: 1.0

☑ **The string documentation of an object (based on docstring)**

☐ Prints the documentation of an object
     (/)
☐ Creates man file

☐ I don't know

9. What is the size of the `int` data type on a 64-bit machine?

Score: 1.0

☐ 1 byte

☐ 2 bytes

☑ **4 bytes**

☐ 8 bytes

☐ I don't know

10. Which of the following sorting algorithms has best case time complexity of `O(nlog(n))` ?

Score: 1.0

☑ **Quick Sort**

☐ Bubble Sort

☐ Insertion Sort

☐ Selection Sort

☐ I don't know

11. What do these lines print?

```
class User:
    id = 1

u = User()
User.id = 98
print(u.id)
```

Score: 1.0

☐ None

- ☐ 1
- ☐ (/)
  89
- ☑ **98**
- ☐ I don't know

12. What do these lines print?

```python
class Base():
    """ My base class """

    __nb_instances = 0

    def __init__(self):
        Base.__nb_instances += 1
        self.id = Base.__nb_instances

class User(Base):
    """ My User class """

    def __init__(self):
        super().__init__()
        self.id = 89

u = User()
print(u.id)
```

Score: 1.0

- ☑ **89**
- ☐ 90
- ☐ 1
- ☐ I don't know

13. What does the following Bash script do?

🔍

```
#!/usr/bin/env bash
    (/)
var="Tech"
if [ -e "$var" ]
then
    if [ -f "$var" ]
    then
        echo "Betty"
    elif [ -d "$var" ]
    then
        echo "School"
    fi
else
    echo "$var doesn't exist"
fi
```

Score: 0.0

☐ **Checks if `Tech` exists, otherwise prints "Tech doesn't exist". If it exists and it's a file, print "Betty", otherwise if it's a directory, print "School".**

☐ Checks if a file (inputted by the user) exists, otherwise prints "File doesn't exist". If it exists and it's a file, print "Betty", otherwise if it's a directory, print "School".

☑ **Checks if `Tech` exists and prints "Tech exists"**

☐ I don't know

14. What do these lines print?

```
>>> class User:
>>>     id = 89
>>>     name = "no name"
>>>     __password = None
>>>
>>>     def __init__(self, new_name=None):
>>>         self.is_new = True
>>>         if new_name is not None:
>>>             self.name = new_name
>>>
>>> u = User()
>>> u.name
```

Score: 1.0

☐ name

☐ None

☐ 'John'

☑ **'no name'**
(/)
☐ I don't know

---

15. What data structure is the foundation of a Python dictionary or set?

Score: 1.0

☑ **Hash Table**
☐ Stack
☐ Queue
☐ Binary Tree
☐ I don't know

16. What is the `unistd` symbolic constant for the standard error?

Score: 0.0

☐ STDIN_FILENO
☑ **STDOUT_FILENO**
☐ **STDERR_FILENO**
☐ I don't know

17. Based on this code, what should all the test cases be?

(select all possible answers)

```
def uniq(list):
    """ Returns unique values of a list """
    u_list = []
    for item in list:
        if item not in u_list:
            u_list.append(item)
    return u_list
```

Score: 1.0

☑ **empty list**
☑ **list with one element (any type)**

☑ **list with 2 different elements (same type)**

(/)

☑ **list with the same element twice (same type)**

☑ **list with more than 2 times the same element (same type)**

☑ **list with multiple types (integer, string, etc...)**

☑ **not a list argument (ex: passing a dictionary to the method)**

☐ I don't know