Preguntas PCAP (Tanda 2)

Which of the following is the output of the below Python code?
list1 = [1, 3]
list2 = list1
list1[0] = 4
print(list2)
○ [1, 3]
<u> </u>
O [4, 3]
O [1, 4]
O [1, 4]
O [1, 3, 4]
∯

Which is the output of the following Python code fragment?
x = True
y=False
z= False
if not x or y:
print (1)
elif not x or not y and z:
print (2)
elif not x or y or not y and x:
print (3)
else:
print (4)
O 4
O 2
О з
None
<u> </u>

What will be the output of the following code?
class Test:
definit(self, s):
self.s = s
def print(self):
print(s)
a = Test("Python Class")
a.print()
The program gives an error because there is no constructor for class Test.
Signature for the print method is incorrect, so an error is thrown.
O The correct output is .
The above code will execute correctly on changing print(s) to print(self.s).
)

0	The program	has an e	rror because	class Tes	t does not	have a	constructor.
---	-------------	----------	--------------	-----------	------------	--------	--------------

- The above code produces an error because the definition of print(s) does not include .
- It executes successfully but prints nothing.

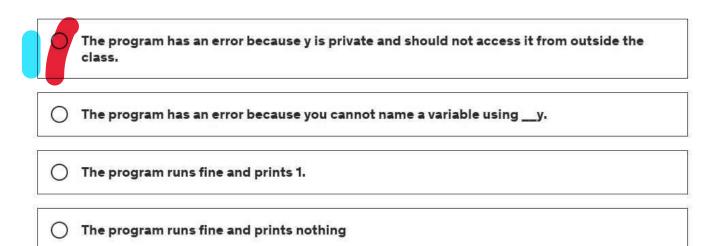
The program has an error because of the constructor call is made without an argument.

0	The program has an error because the constructor is not present in class Test.
0	The above code produces an error because the definition of print(s) does not contain .
0	It executes successfully but prints nothing.
0	The program has an error because of the constructor call is made without an argument.



The program executes successfully and prints Welcome.

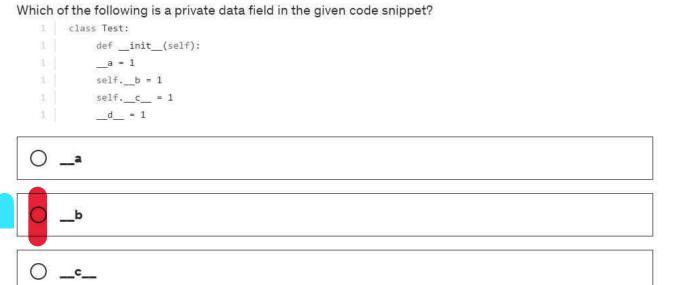
- The program has an error because x is private and cannot be accessed outside of the class.
- O The program has an error because you cannot name a variable using .
- The program runs fine and prints 1.
 - The program runs fine and prints nothing.



0	The program has an error because x is private and should not access it from outside the
	class.

- The program has an error because you cannot name a variable using __y.
- O The program runs fine and prints 1.





O _d_

What	will be the output of the following code snippet?
1	class Test:
1	<pre>definit(self):</pre>
1	self.x = 1
1	selfy = 1
1	
1	<pre>def getY(self);</pre>
1	return selfy
1	
1	<pre>val= Test()</pre>
1	valy = 45
1	<pre>print(val.getY())</pre>
0	The program has an error because y is private and should not access it from outside the class.
0	The program has an error because you cannot name a variable usingy.
T .	
0	The code runs fine and prints 1.
1	
0	The code executes successfully and prints 45.

```
def main():
    myCounter = Counter()
    num = 0

for i in range(0, 100):
    increment(myCounter, num)

print("myCounter.counter =", myCounter.counter, ", number of times =", num)

def increment(c, num):
    c.counter += 1
    num += 1

class Counter:
    def __init__(self):
        self.counter = 0

main()
```

- ounter is 101, number of times is 0
- counter is 100, number of times is 0
- ounter is 100, number of times is 100
- ounter is 101, number of times is 101

What code can we put at the third line of the definition of class B to invoke its superclass's constructor?



```
class A:
    def __init__(self, x = 1):
        self.x = x

class B(A):
    def __init__(self, y = 2):
        super().__init__()
        self.y = y

def main():
    b = B()
    print(b.x, b.y)

main()
```

0	0
0	1
0	2
0	2

O 110			
O 120			
O 210			
O 220			

```
1 class A:
      def __init__(self, x = 0):
           self.x = x
      def func1(self):
           self.x += 1
  class B(A):
       def __init__(self, y = 0):
          A.__init__(self, 3)
          self.y = y
       def func1(self):
           self.y += 1
   def main():
       b = B()
       b.func1()
       print(b.x, b.y)
   main()
```

O 30

O 20		
O 31		
O 40		

What will be the output of the following code snippet? What will be the output of the following code snippet?

```
class A:
        def __new__(self):
           self.__init__(self)
           print("A's __new__() invoked")
       def __init__(self):
            print("A's __init__() invoked")
2
    class B(A):
       def __new__(self):
           print("B's __new__() invoked")
      def __init__(self):
           print("B's __init__() invoked")
    def main():
        b = B()
        a = A()
    main()
```

A's __init__() invoked A's __new__() invoked

B's __new__() invoked A's __init__() invoked

B's __new__() invoked A's __new__() invoked

B's __new__() invoked A's __init__() invoked A's __new__() invoked

```
class A:
    def __init__(self, num):
        self.x = num

class B(A):
    def __init__(self, num):
        self.y = num

obj = B(11)
    print ("%d %d" % (obj.x, obj.y))
```

O None 11	
11 None	
O 1111	
AttributeError: 'B' object has no attribute 'x'	

x is not accessible from the object of classB.

```
class A:
    def __init__(self):
        self.x = 1

def func(self):
        self.x = 10

class B(A):
    def func(self):
        self.x += 1
        return self.x

def main():
    b = B()
    print(b.func())

main()
```

O 1	
	21
O 2	8
O 10	

○ ABC

О СВА

O AAA

```
class A:
    def __init__(self, x = 2, y = 3):
        self.x = x
        self.y = y

def __str__(self):
        return "A"

def __eq__(self, num ):
        return self.x * self.y == num.x * num.y

def main():
        a = A(1, 2)
        b = A(2, 1)
        print(a == b)

main()
```

	*
0	True
0	False
0	2