Your score: **20/20**

100%

Congratulations, you've passed the test!

SECTION ANALYSIS

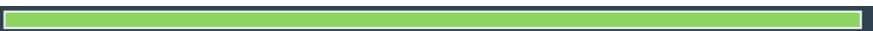| PE1 -- Module 3 Test	100%

☰ Question 1/20 Next ➜

How many stars (*) will the following snippet send to the console?

```
i = 0
while i <= 3 :
    i += 2
    print("*")
```
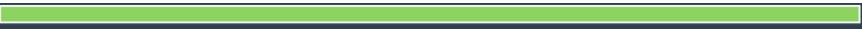
- ○ one
- ○ three
- ○ zero
- ● two

☰ Question 2/20 ← Prev Next ➜

What is the output of the following snippet?

```
a = 1
b = 0
c = a & b
d = a | b
e = a ^ b

print(c + d + e)
```

- ● 2
- ○ 3
- ○ 1
- ○ 0

☰   Question 3/20   ← Prev   Next →

🏳 🗑

What is the output of the following snippet?

```
my_list_1 = [1, 2, 3]
my_list_2 = []
for v in my_list_1:
    my_list_2.insert(0, v)
print(my_list_2)
```

- ⦿ `[3, 2, 1]`

- ○ `[1, 1, 1]`

- ○ `[3, 3, 3]`

- ○ `[1, 2, 3]`

☰   Question 4/20   ← Prev   Next →

🏳 🗑

What value will be assigned to the `x` variable?

```
z = 10
y = 0
x = y < z and z > y or y > z and z < y
```

- ○ `False`

- ⦿ `True`

- ○ `1`

- ○ `0`

☰   Question 5/20   ← Prev   Next →

🏳 🗑

How many hashes (#) will the following snippet send to the console?
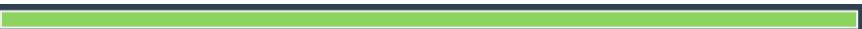
```
var = 0
while var < 6:
    var += 1
    if var % 2 == 0:
        continue
    print("#")
```

- ○ one

- ⦿ three

- ○ zero

- ○ two

☰   Question 6/20   ← Prev   Next →

🏳 🗑

What is the output of the following snippet?

```
my_list = [1, 2, 3, 4]
print(my_list[-3:-2])
```

- ○ `[2, 3]`

- ○ `[2, 3, 4]`

- ⦿ `[2]`

- ○ `[]`

☰   Question 7/20   ← Prev   Next →

🏳 🗑

The second assignment:

```
vals = [0, 1, 2]
vals[0], vals[2] = vals[2], vals[0]
```

- ○ shortens the list

- ○ extends the list

- ○ doesn't change the list

- ⦿ reverses the list

☰   Question 8/20   ← Prev   Next →

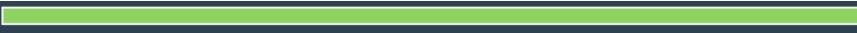Which of the following sentences are true? (Select two answers)

```
nums = [1, 2, 3]
vals = nums[-1:-2]
```

- ☑ `nums` is longer than `vals`
- ☐ `vals` is longer than `nums`
- ☐ `nums` and `vals` are of the same length
- ☑ `nums` and `vals` are two different lists

☰   Question 9/20   ← Prev   Next →

How many hashes (#) will the following snippet send to the console?
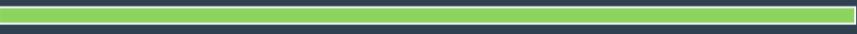
```
var = 1
while var < 10:
    print("#")
    var = var << 1
```

- ◯ one
- ◯ two
- ◯ eight
- ⦿ four

☰   Question 10/20   ← Prev   Next →

What is the output of the following snippet?
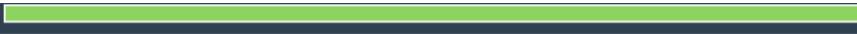
```
my_list = [3, 1, -2]
print(my_list[my_list[-1]])
```

- ⦿ `1`
- ◯ `-2`
- ◯ `-1`
- ◯ `3`

☰   Question 11/20   ← Prev   Next →

Take a look at the snippet, and choose the true statements: (Select two answers)

```
nums = [1, 2, 3]
vals = nums
del vals[1:2]
```

- ☐ `nums` is replicated and assigned to `vals`
- ☑ `nums` and `vals` are of the same length
- ☑ `nums` and `vals` refer to the same list
- ☐ `nums` is longer than `vals`

☰   Question 12/20   ← Prev   Next →

How many hashes (#) will the following snippet send to the console?

```
for i in range(1):
    print("#")
else:
    print("#")
```

- ◯ three
- ◯ zero
- ◯ one
- ⦿ two

☰ Question 13/20 ← Prev | Next →

What is the output of the following snippet?

```
t = [[3-i for i in range (3)] for j in range (3)]
s = 0
for i in range(3):
    s += t[i][i]
print(s)
```
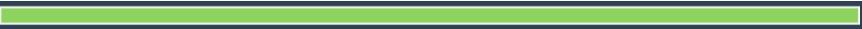
- ○ 02
- ● 6
- ○ 7
- ○ 4

☰ Question 14/20 ← Prev | Next →

How many stars (*) will the following snippet send to the console?

```
i = 0
while i <= 5 :
    i += 1
    if i % 2 == 0:
        break
    print("*")
```

- ○ three
- ● one
- ○ zero
- ○ two

☰ Question 15/20 ← Prev | Next →

What is the output of the following snippet?

```
my_list = [[0, 1, 2, 3] for i in range(2)]
print(my_list[2][0])
```

- ● the snippet will cause a runtime error
- ○ 0
- ○ 1
- ○ 2

☰ Question 16/20 ← Prev | Next →

After execution of the following snippet, the sum of all `vals` elements will be equal to:

```
vals = [0, 1, 2]
vals.insert(0, 1)
del vals[1]
```

- ○ 2
- ● 4
- ○ 3
- ○ 5

☰ Question 17/20 ← Prev | Next →

How many elements does the `my_list` list contain?

```
my_list = [i for i in range(-1, 2)]
```

- ○ four
- ○ one
- ○ two
- ● three

≡ Question 18/20 ← Prev Next →

The value eventually assigned to `x` is equal to:

```
x = 1
x = x == x
```

- ○ `1`
- ○ `False`
- ● `True`
- ○ `0`

≡ Question 19/20 ← Prev Next →

What is the output of the following snippet?

```
my_list = [1, 2, 3]
for v in range(len(my_list)):
    my_list.insert(1, my_list[v])
print(my_list)
```

- ● `[1, 1, 1, 1, 2, 3]`
- ○ `[1, 2, 3, 3, 2, 1]`
- ○ `[1, 2, 3, 1, 2, 3]`
- ○ `[3, 2, 1, 1, 2, 3]`

≡ Question 20/20 ← Prev

An operator able to check whether two values are equal is coded as:

- ○ `===`
- ○ `!=`
- ○ `=`
- ● `==`