1 True or False 1 point

In Python the bodies of if/else statements, for loops, and while loops need to be properly and consistently indented.

- (●) True
- ( ) False

2 Multiple Choice 1 point

What statement best describes the relationship between if statements and while loops?

- ( ) while loops never contain if statements
- ( ) while loops always contain at least one if statement
- (●) It is not uncommon for while loops to contain if statements
- ( ) while loops may contain if statements, but it almost never makes sense to do so

3 Multiple Choice 1 point

Which of the following statements is true about “else” and “elif” in the context of Python control structures?

- (●) “elif” needs a condition, whereas “else” does not
- ( ) “elif” and “else” are both used as catch-alls after a sequence of if-statements where all conditions are not met
- ( ) “elif” is a general catch-all whereas “else” is not
- ( ) “elif” and “else” cannot show up together in a code segment used for conditional processing

4 Multiple Choice 1 point

Review the following code snippet. What is the output?

```
var1 = 1
while var1 < 10:
    var1 = var1 * 2
    if var1 == 4:
        continue
    if var1 == 6:
        break
print(var1)
```

- ○ 2
- ○ 4
- ○ 6
- ● 16

5 True or False 1 point

Generally, a for-loop is used when the number of iterations is known ahead of time (e.g. iterating over an iterable like a list of items or iterating a specific number of times) whereas a while-loop will iterate until a particular condition is met and it might not be apparent how many iterations will occur. Is this statement true or false?

- ● True
- ○ False

6 Multiple Choice 1 point

What is the output of the following code, if $n = 10345$?

```
length = 0
while n > 0:
    n //= 10 # this is equivalent to n = n // 10
    length += 1
print(length)
```

- ○ 0
- ○ 3
- ○ 4
- ● 5

7 True or False 1 point

An iterator knows when it is at the end of the iterable object and therefore, you can call the next() method (which gets the next item from the iterator) as many times as you want, and it will not be a problem since it will always stop once it runs out of items to retrieve.

- ○ True
- ● False

8 Multiple Choice 1 point

What does the following code print?

```
powers = [2,4,8,16,32,64]
some_vals = [int(x/2) for x in powers if x < 10]
print(some_vals)
```

- [ ] `[2, 4, 8]`
- [x] `[1, 2, 4]`
- [ ] `[1, 2, 4, 8, 16, 32]`
- [ ] none of these

9 Multiple Choice 1 point

What does the following code print?

```
codedMessage = '.GERYEXAKTL!'
decodedMessage = [char for index, char in enumerate(codedMessage)
    if (index % 2 == 1)]
print(decodedMessage)
```

- `['.','G','E','R','Y','E','X','A','K','T','L','!']`
- `['G']`
- `['G', 'R', 'E', 'A', 'T', '!']`
- `['.', 'E', 'Y', 'X', 'K', 'L']`
- `['G', 'R', 'E', 'Y', '!']`

10 Multiple Answer 1 point

Which of these are true statements about list comprehensions? Select all that apply:

- [x] they produce a list
- [ ] they must contain a while-loop
- [x] they must contain a for-loop
- [x] they may contain one or more if-statements