1. Python is an interpreted language. Which of the following statements correctly describes an interpreted language? **1 / 1 point**

- ○ The source code is pre-built and compiled before running.
- ◉ The source code is converted into bytecode that is then executed by the Python virtual machine.
- ○ Python will save all code first prior to running.
- ○ Python needs to be built prior to it being run.

**Correct**
Correct! Unlink other programming languages Python does not need to be built or linked for the code to run.

2. Why is indentation important in Python? **1 / 1 point**

- ○ The code will compile faster with indentation.
- ◉ Python used indentation to determine which code block starts and ends.
- ○ It makes the code more readable.
- ○ The code will be read in a sequential manner

**Correct**
Correct! Python does not use curly braces like other languages, so it leverages off indentation to determine where the code blocks start and end.

**3.** What will be the output of the following code?

1 / 1 point

```
names = ["Anna", "Natasha", "Mike"]
names.insert(2, "Xi")
print(names)
```

- ["Anna", "Natasha", 2, "Xi", "Mike"]
- ["Anna", "Xi", "Mike" ]
- ["Anna", "Natasha", "Xi", "Mike"] ✓
- ["Anna", "Natasha", Xi]

**Correct**
Correct! The insert() function displaces the remaining list after inserting the element passed.

**4.** What will be the output of the code below?

1 / 1 point

```
for x in range(1, 4):
    print(int((str((float(x))))))
```

- 1.0, 2.0
- "one", "two"
- Will give an error ✓
- 1 , 2

**Correct**
Correct! The float will first convert into string and output such as <class 'float'> which cannot be converted into int.

**5.** What will be the output of the following code:

1 / 1 point

```
sample_dict = {1: 'Coffee', 2: 'Tea', 3: 'Juice'}
for x in sample_dict:
    print(x)
```

- (●) 1 2 3
- ( ) {1 2 3}
- ( ) 'Coffee', 'Tea', 'Juice'
- ( ) (1, 'Coffee')

  (2, 'Tea')

  (3, 'Juice')

**Correct**

Correct! The default values printed from a dictionary are keys.

**6.** What will be the output of the recursive code below?

1 / 1 point

```
def recursion(num):
    print(num)
    next = num - 3
    if next > 1:
        recursion(next)

recursion(11)
```

- 2 5 8 11
- 2 5 8
- 11 8 5 2 (selected)
- 8 5 2

**Correct**
Correct! The values printed have difference of 3, but printed in opposite order.

**7.** What will be the type of time complexity for the following piece of code:

1 / 1 point

```
def bigo(numbers):
    for i in numbers:
        print(numbers)

bigo([1, 7, 13, 19])
```

- Logarithmic Time
- Constant Time
- Quadratic Time
- Linear Time (selected)

**Correct**
Correct! The single for loop will have linear time depending on the size of the input sequence.

**8.** What will be the output of the code below:

1 / 1 point

```
str = 'Pomodoro'
for l in str:
if l == 'o':
    str = str.split()
    print(str, end=", ")
```

- (●) Will throw an error
- ( ) ['Pomodoro']
- ( ) ['P', 'm', 'd', 'o']
- ( ) ['Pomodoro', 'modoro', 'doro', 'ro']

**Correct**
Correct! The first time split() function is used, the str variable will convert into a list over which split() cannot be used and will give an error.

---

**9.** Find the output of the code below:

1 / 1 point

```
def d():
    color = "green"
    def e():
        nonlocal color
        color = "yellow"
    e()
    print("Color: " + color)
    color = "red"
color = "blue"
d()
```

- (●) yellow
- ( ) blue
- ( ) green
- ( ) red

**Correct**
Correct! The color variable will retain the value from the nonlocal variable from e()

**10.** Find the output of the code below:

1 / 1 point

```
num = 9
class Car:
    num = 5
    bathrooms = 2

def cost_evaluation(num):
    num = 10
    return num

class Bike():
    num = 11

cost_evaluation(num)
car = Car()
bike = Bike()
car.num = 7
Car.num = 2
print(num)
```

- ○ 5
- ○ 2
- ◉ 9
- ○ 10

**Correct**
Correct! The value of the global variable will remain unchanged.

**11.** Which of the following is the correct implementation that will return **True** if there is a parent class P, with an object p and a sub-class called C, with an object c?

1 / 1 point

- ○ print(issubclass(C,c))
- ○ print(issubclass(p,C))
- ○ print(issubclass(P,C))
- ◉ print(issubclass(C,P))

**Correct**
Correct! It can be read as C is sub-class of P.

**12.** Django is a type of: **1 / 1 point**

- (•) Full-stack framework
- ( ) Micro-framework
- ( ) Asynchronous framework

**Correct**
Correct! Django is a Full-stack framework.

**13.** Which of the following is not true about Integration testing: **1 / 1 point**

- ( ) Primarily dealt by the tester.
- ( ) It combines unit tests.
- (•) It is where the application is tested as a whole.
- ( ) Tests the flow of data from one component to another.

**Correct**
Correct! This is the case with system testing.

**14.** While using pytest for testing, it is necessary to run the file containing the main code before we can run the testing file containing our unit tests. **1 / 1 point**

- (•) False
- ( ) True

**Correct**
Correct! The main file must be saved to keep it updated but it is not required to be executed. We have to import it into our testing file.

**15.** What will be the output of the code below:

1 / 1 point

```
class A:
    def a(self):
        return "Function inside A"

class B:
    def a(self):
        return "Function inside B"

class C:
    pass

class D(C, A, B):
    pass

d = D()
print(d.a())
```

- (●) Function inside A
- ( ) Function inside B
- ( ) None of the above
- ( ) No output

**Correct**

Correct! The class A comes before class B in terms of the parent classes of class D.