Q1. How can we create an iterator object from a list?

a) Bypassing the given list to the iter() function

b) By using a for a loop.

c) By using a while loop.

d) You cannot create an iterable object

Q2. If the function contains at least of one “yield” statement, then it becomes ______

Choose one

a) An iterable

b) a generator function

c) an anonymous function

d) None of the above

Q3. What is the output of the code?

```
mylist = [1, 3, 6, 10]
a = (x**2 for x in mylist)
print(next(a), next(a))
```

a) 1 3

b) 1 9

c) 1936100

d) 1

Q4. What are the criteria that must be met to create closure in Python?

a) The program Must have the function inside the function.

b) The nested function must refer to the value defined in the enclosing function.

c) The enclosing function must return the nested

d) All of the above.

Q5. What is the output of the code?

```
def Foo(n):
    def multiplier(x):
        return x * n
    return multiplier

a = Foo(5)
b = Foo(5)

print(a(b(2)))
```

a) 25.

b) 100

c) 10

d) 50

Q6. What is the output of the code?

```
def make_pretty(func):
    def inner():
        print("I got decorated")
        func()
    return inner

def ordinary():
    print("I am ordinary")

pretty = make_pretty(ordinary)
pretty()
```

a) I got decorated

b) I am pretty

c) I got decorated I am ordinary

d) am ordinary I got decorated

Q7: What is the more pythonic way to use getters and setters?

a) Decorators

b) Generators.

c) Iterators

d) @property

Q8. In Python, there is a built-in function property() that returns a property object. The property object has which of the methods?

a) getter() and setter()

b) getter(), setter() and delete()

c) getter() and delete()

d) setter() and delete()

Q9. Which of the following statement is true?

a) You cannot chain multiple decorators in Python.

b) Decorators don't work with functions that take parameters.

c) The @ symbol doesn't have any use while using decorators.

d) None of the above

Q10. For the following codes, which of the following statements is true?

```
def printHello():
   print("Hello")
a = printHello()
```

a) Print Hello() is a function, and a is a variable. None of them are objects.

b) Both printHello() and the reference to the same object.

c) Print Hello() and the reference to different objects.

d) Syntax error! You cannot assign function

Q11. What is the output of the program?

```
def outerFunction():
   global a
   a = 20
   def innerFunction():
       global a
       a = 30
       print('a =', a)
a = 10
outerFunction()
print('a =', a)
```

a) a = 10 a = 30

b) a = 10

c) a = 2

d) a = 30

Q12. Which of the following statements is true?

a) A class is a blueprint for the object.

b) You can only make the single object from the given class

c) Both statements are true.

d) Neither statement is true.

Q13. What is the output of the code?

```
class Foo:
  def printLine(self, line='Python'):
    print(line)

o1 = Foo()
o1.printLine('Java')
```

a) Python

b) Line

c) Java

d) Java
Python

Q14. What is the function of the `__init__()` function in Python?

a) Initialises the class for use.

b) This function is called, when the new object is instantiated

c) Initialises all the data attributes to zero when called

d) None of the above.

Q15. What is the output of the code?

```
class Point:
    def __init__(self, x = 0, y = 0):
      self.x = x+1
      self.y = y+1

p1 = Point()
print(p1.x, p1.y)
```

a) 0 0

b) 1 1

c) None None

d) x y

Q16. Which of the following code used the inheritance feature?

a)

```
Class Foo:
Pass
```

b)

```
class Foo(object):
pass

    class
Hoo(object): pass
```

c)

```
class Foo:
pass

class
Hoo(Foo): pass
```

d) None of the above code.

Q17 If you a class is derived from two different classes, it’s called ________

a. Multilevel inheritance

b. Multiple Inheritance

c. Hierarchical Inheritance

a. Python Inheritance