Python Basics

Part of assignment #0

Describe what each snippet of code does or prints. Write a sentence why.

Name: _____

1. What does the following code do and why?

```
def a(b, c, d):
    pass
```

2. What is printed by the following code? Why?

```
print(type([1,2]))
```

3. What is printed by the following code? Why?

```
def f():
    pass

print(type(f()))
```

4. What is printed by the following code? Why?

```
print(type(1J))
```

5. What is printed by the following code? Why?

```
print(type(lambda:None))
```

6. What is printed by the following code? Why?

```
a = [1,2,3,None,(),[],]
print(len(a))
```

7. What is printed by the following code (in Python 3)?

```
print(type(1/2))
```

8. What is printed by the following code? Why?

```
d = lambda p: p * 2
t = lambda p: p * 3
x = 2
x = d(x)
x = t(x)
x = d(x)
print(x)
```

9. What is printed by the following code? Why?

```
x = 4.5
y = 2
print(x//y)
```

10. What is printed by the following code? Why?

```
nums = set([1,1,2,3,3,3,4])
print(len(nums))
```

11. What is printed by the following code? Why?

```
x = True
y = False
z = False

if x or y and z:
    print("yes")
else:
    print("no")
```

12. What is printed by the following code? Why?

```
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)
```

13. What is printed by the following code? Why?

```
print(r"\nwoow")
```

14. What is printed by the following code? Why?

```
class parent:
    def __init__(self, param):
        self.v1 = param

class child(parent):
    def __init__(self, param):
        self.v2 = param

obj = child(11)
print(obj.v1 + " " + obj.v2)
```

15. What is printed by the following code? Why?

```
class Account:
    def __init__(self, id):
        self.id = id
        id = 999

acc = Account(123)
print(acc.id)
```

16. What is printed by the following code? Why?

```
name = "snow storm"

print(name[6:8])
```

17. What is printed by the following code? Why?

```
name = "snow storm"
name[5] = 'X'
print(name)
```

18. What is printed by the following code? Why?

```
for i in range(2):
    print(i)

for i in range(4,6):
    print(i)
```

19. What is printed by the following code? Why?

```
values = [2, 3, 2, 4]

def my_transformation(num):
    return num ** 2

for i in  map(my_transformation, values):
    print(i)
```

20. What is printed by the following code? Why?

```
import math
print(math.floor(5.5))
```

21. What is printed by the following code? Why?

```
x = "foo "
y = 2
print(x + y)
```

22. Which piece of code will print all of the names in the list on a new, separate line?

```
names = ['Ramesh', 'Rajesh', 'Rachel', 'Eileen', 'Nico']
```

23. Assuming the filename for the code below is /usr/lib/python/person.py and the program is run as:
`python /usr/lib/python/person.py`

What get's printed?

```
class Person:
    def __init__(self):
        pass

    def getAge(self):
        print(__name__)

p = Person()
p.getAge()
```

24. What is printed by the following code? Why?

```
foo = {}
print(type(foo))
```

25. What is printed by the following code? Why?

```
foo = (3, 4, 5)
print(type(foo))
```

26. What is printed by the following code? Why?

```
country_counter = {}

def addone(country):
    if country in country_counter:
        country_counter[country] += 1
    else:
        country_counter[country] = 1

addone('China')
addone('Japan')
addone('china')

print(len(country_counter))
```

27. What is printed by the following code? Why?

```
confusion = {}
confusion[1] = 1
confusion['1'] = 2
confusion[1] += 1

total = 0
for k in confusion:
    total += confusion[k]

print(total)
```

28. What is printed by the following code? Why?

```
confusion = {}
confusion[1] = 1
confusion['1'] = 2
confusion[1.0] = 4

total = 0
for k in confusion:
    total += confusion[k]

print(total)
```

29. What is printed by the following code? Why?

```
boxes = {}
jars = {}
crates = {}

boxes['cereal'] = 1
boxes['candy'] = 2
jars['honey'] = 4
crates['boxes'] = boxes
crates['jars'] = jars

print(len(crates[boxes]))
```

30. What is printed by the following code? Why?

```
numberGames = {}
numberGames[(1,2,4)] = 8
numberGames[(4,2,1)] = 10
numberGames[(1,2)] = 12

total = 0
for k in numberGames:
    total += numberGames[k]

print(len(numberGames) + total)
```

31. What is printed by the following code? Why?

```
foo = {1:'1', 2:'2', 3:'3'}
foo = {}
print(len(foo))
```

32. What is printed by the following code? Why?

```
foo = {1:'1', 2:'2', 3:'3'}
del foo[1]
foo[1] = '10'
del foo[2]
print(len(foo))
```

33. What is printed by the following code? Why?

```
names = ['Amir', 'Barry', 'Chales', 'Dao']
print(names[-1][-1])
```

34. What is printed by the following code? Why?

```
names1 = ['Amir', 'Barry', 'Chales', 'Dao']

if 'amir' in names1:
    print(1)
else:
    print(2)
```

35. What is printed by the following code? Why?

```
names1 = ['Amir', 'Barry', 'Chales', 'Dao']
names2 = [name.lower() for name in names1]

print(names2[2][0])
```

36. What is printed by the following code? Why?

```
numbers = [1, 2, 3, 4]

numbers.append([5,6,7,8])

print(len(numbers))
```

37. What is printed by the following code? Why?

```
list1 = [1, 2, 3, 4]
list2 = [5, 6, 7, 8]

print(len(list1 + list2))
```

38. What is printed by the following code? Why?

```
a = 1
b = 2
a,b = b,a

output = "{} {}".format(a, b)
print(output)
```

39. What is printed by the following code? Why?

```
def myfunc(x, y, z, a):
    print(x + y)

nums = [1, 2, 3, 4]

myfunc(*nums)
```

40. What is printed by the following code? Why?

```
import numpy as np
ary = np.array([1,2,3,5,8])
ary = ary + 1
print (ary[1])
```

41. What is printed by the following code? Why?

```
import numpy as np

a = np.array([1,2,3,5,8])
b = np.array([0,3,4,2,1])
c = a + b
c = c*a

print (c[2])
```

42. What is printed by the following code? Why?

```
import numpy as np
a = np.array([1,2,3,5,8])
print (a.ndim)
```

43. What is printed by the following code? Why?

```
import numpy as np
a = np.array([[1,2,3],[0,1,4]])
print (a.size)
```

44. What is printed by the following code? Why?

```
import numpy as np

a = np.array([[0, 1, 2], [3, 4, 5]])
b = a.sum(axis=1)
print (b)
```

45. What is printed by the following code? Why?

```
import numpy as np

a = np.array([[1, 2, 3], [4, 5, 6]])
a += 3
print(a[1,2])
```