Python test

Part 1 – Programming:

1. Write a program (with function) which takes a sequence of numbers and check if all numbers are unique.
2. Python program to find out the average of a set of integers
    a. Note: Don't use the pre-defined functions like np.mean
    b. For ex: If input is 5, the output should be average of 1+2+3+4+5
3. Python program to check whether the given integer is a multiple of both 5 and 7
4. Python program to display the given integer in reverse manner
    a. Ex: input 852, output should be 258
5. Create an inner function to calculate the addition in the following way
    a. Create an outer function that will accept two parameters, a and b
    b. Create an inner function inside an outer function that will calculate the addition of a and b
    c. At last, an outer function will add 5 into addition and return it
6. Python Program to Check Leap Year
    a. A leap year is exactly divisible by 4 except for century years (years ending with 00). The century year is a leap year only if it is perfectly divisible by 400
    b. For ex: 2017 is not a leap year, 1900 is a not leap year, 2012 is a leap year, 2000 is a leap year

Part 2 – Pandas exercise (basic)

Note: Use automobile dataset for below questions

1. Find the most expensive car company name
2. Count total cars per company
3. Find each company's highest price car
4. Sort all cars by Price column
5. Group by make, fuel-type and body-style to calculate the average of city-mpg and highway-mpg (miles per gallon)

Part 3 - MCQ:

1

Which of the following lines of code generate the same "result" value as the Python statements below?

```
data = [1,2,3,4,5,6]

def f1(x):
    return 3 * x

def f2(x):
    try:
        return x > 3
    except:
        return 0


result = list(map(f1, list(filter(f2, data))))
```

- ☐ `result = [3*i for i in data(i) if i > 3]`
- ☐ `result = list(map(lambda r: r * 3, filter(f2, data)))`
- ☐ `result = list(map(lambda r: r * 3, filter(lambda f: f>3 or 0, data)))`
- ☐ `result = [3*i for i in data if i > 3]`
- ☐ `result = [3*i for i in data if i > 3 else 0]`

2

Which of the following expressions describe Python identity operators?

- ☐ `is is not`
- ☐ `not is`
- ☐ `not is is`
- ☐ `is not`
- ☐ `is not is`

3.

Given a Python function `foo()` that will raise an exception, which of following code snippets will be used to assign the exception to a variable and print it out?

- [ ]
```
try:
    foo()
except BaseException:
    print(BaseException)
```

- [ ]
```
try:
    foo()
except BaseException:
    print(__error__)
```

- [ ]
```
try:
    foo()
except BaseException as variable:
    print(variable)
```

- [ ]
```
try:
    foo()
except BaseException, variable:
    print(variable)
```

- [ ]
```
try:
    foo()
except BaseException:
    print(this.exception)
```

4.

Which of following lines of Python code correctly format the value supplied as input and provide an output of "3.14"?

- [ ] `print("{:.2f}".format(**pi())")`
- [ ] `print("%.2f" % 3.142)`
- [ ] `print("{.2f}".format(3.142))`
- [ ] `print("{:.2f}".format(3.142))`
- [ ] `print("{n:.2f}".format(n=3.142))`

5.

If a list called `vector` contains `[[1,2],[3,4],[5,6],[7,8],[9,10]]`, which of the following will be produced by the Python statement below?

```
( x for y in vector for x in y if x % 2 == 0 )
```

- ☐ A list containing the integer 10
- ☐ A list containing two lists, one with 3 and 4, the other one with 7 and 8
- ☐ A list containing all integers from `vector`
- ☐ A generator which will generate single integer lists with all even integers from `vector`
- ☐ A generator which will generate all even integers from `vector`

6.

After running the Python program below, which of the following correctly identify the final value of the `Z` variable?

```
Z = 5
X = "456"
while Z > -1:
    Y = X
    if Y in ('1', '2', '3') or len(Y) > 1:
        break
    X = X + Y
    if Y in ('4', '5', '6'):
        continue
    Z = Z - 1
print(Z)
```

- ☐ 3
- ☐ 5
- ☐ 4
- ☐ -1
- ☐ 2

7.

Which of the following Python statements will return a tuple of two values from a function?

- [ ] `return Pair(x,y)`
- [ ] `return new Pair(x,y)`
- [ ] `return x,y`
- [x] `return (x,y)`
- [ ] `return {"x":x, "y":y}`

8.

After running the Python code below, which of the following correctly describe the values of `before` and `after`?

```
before = [1,2,3,4,5]

def bar(lst) :
    return lst.append(7)

after = bar(before)
```

- [ ] `before = [1,2,3,4,5,7]`
  `after = None`
- [x] `before = [1,2,3,4,5,7]`
  `after = [1,2,3,4,5,7]`
- [ ] `before = [1,2,3,4,5]`
  `after = [1,2,3,4,5,7]`
- [ ] `before = [1,2,3,4,5]`
  `after = [1,2,3,4,5]`
- [ ] `before = [1,2,3,4,5]`
  `after = None`

9.

Which of the following messages will be printed as a result of executing the Python program below?

```
def f():
    yield True

try:
    g = f()
    h = next(g)
    assert h is True
    print("True")
    h = next(g)
    assert h is None
    print("None")
except AssertionError:
    print("Assertion failed")
except GeneratorExit:
    print("Exit")
except StopIteration:
    print("Stop")
```

- ☐ None
- ☐ Exit
- ☐ True
- ☐ Stop
- ☐ Assertion failed

10.

A developer is using the Python code below to test whether an application is working. Which of the following correctly describe the output of executing this code?

```
i = 150
j = 300

if ((True == False) and (False in (False,))) == True:
    print(i)
elif (True == False) in (False,) == False:
    print(j)
else :
    print("No arithmetical operator proceeded")
```

- ☐ No arithmetical operator proceeded
- ☐ 300
- ☐ i
- ☐ 150
- ☐ j

11.

A list of numbers must be sorted in ascending order by their absolute values.

For example, `[-1,-4, 2, 5, -3]` must be sorted as `[-1, 2, -3, -4, 5]`.

Which of the following Python expressions will accomplish this, correctly sorting `lst` and assigning the result to the variable `slst`?

- ☐ `slst = sorted(lst)`
- ☐ `slst = [x[1] for x in sorted([(abs(x),x) for x in lst])]`
- ☐ `slst = sorted(lst, key=abs)`
- ☐ `slst = sort(lst)`
- ☐ `slst = sort(lst, key=abs)`

12.

Which of the following options are printed by the Python program below?

```
from string import *
method="METHODS"
def x(methods):
    method = str.swapcase(methods)
    print(("%(method)s" % locals()))
methods=str.swapcase(method[:-1])
x(methods)
```

- ☐ method
- ☐ Method
- ☐ METHOD
- ☐ METHODS
- ☐ methods

13.

Which of the following correctly substitute ***** in the Python code snippet below to produce the output below?

Code:

```
class Example:
    def example(self):
        return self.test()

    def test(self):
        return 'Example'

class Test(Example):
    def test(self):
        return 'Test'

*****
```

Output:

```
>>> Example Test
```

Output:

```
>>> Example Test
```

- ☐
```
print(Example().test(),Test().test())
```
- ☐
```
Example1=Example()
Example2=Test()
print(Example1.example(),Example2.test())
```
- ☐
```
print(Example.test(),Test.test())
```
- ☐
```
Example1=Example()
Example2=Test()
print(Example1.test(),Example2.test())
```
- ☐
```
print(test.Example(), test.Test())
```