**Ahmed Hesham Mohamed**

Question 1: 1) the list is a type of container in Python Data Structures, which is used to store multiple data at the same time and can be created using square brackets. Lists are a useful tool for preserving a sequence of data and further iterating over it.

A Tuple is also a sequence data type that can contain elements of different data types, but these are immutable in nature. tuples also can store multiple items in a single variable and can be declared using parentheses. The tuple is faster than the list because of static in nature.

Tuples are immutable, whereas lists are mutable, and this is the main distinction between the two. The values in a list can be changed or modified, while the values of a tuple cannot.

2)The loop control statements break the flow of execution and terminate/skip the iteration as per our need. Python break and continue are used inside the loop to change the flow of the loop from its standard procedure.

The break statement takes care of terminating the loop in which it is used. If the break statement is used inside nested loops, the current loop is terminated, and the flow will continue with the code followed that comes after the loop.

The continue statement skips the code that comes after it, and the control is passed back to the start for the next iteration.

Python pass is a null statement. When the Python interpreter comes across the across pass statement, it does nothing and is ignored.

3) Whenever we create a class in Python, the programmer needs a way to access its attributes and methods. In most languages, there is a fixed syntax assigned to refer to attributes and methods; for example, C++ uses **(this)** for reference.

In Python, the word ***self*** is the first parameter of methods that represents the instance of the class. Therefore, in order to call attributes and methods of a class, the programmer needs to use self.

4) A docstring is simply a multi-line string, that is not assigned to anything. It is specified in source code that is used to document a specific segment of code. Unlike conventional source code comments, the docstring should describe what the function does, not how.

5) Python allows a class to inherit from multiple classes. If a class inherits from two or more classes, you’ll have multiple inheritance.

To extend multiple classes, you specify the parent classes inside the parentheses () after the class name of the child class like this:

class ChildClass(ParentClass1, ParentClass2, ParentClass3):

Question 2:

1. True
2. True
3. True
4. False
5. False

Question 5:

1)

Default constructor

When we do not declare a constructor

In this example, we do not have a constructor but still we are able to create an object for the class. This is because there is a default constructor implicitly injected by python during program compilation, this is an empty default constructor that looks like this:

def \_\_init\_\_(self):

# no body, does nothing.

Parameterized constructor example

When we declare a constructor in such a way that it accepts the arguments during object creation then such type of constructors are known as Parameterized constructors. As you can see that with such type of constructors we can pass the values (data) during object creation, which is used by the constructor to initialize the instance members of that object.

# parameterized constructor

def \_\_init\_\_(self, data):

self.num = data

2)

An object is an instantiation of a class.

Think of a class like the blueprint of a car.

Ford make cars (objects) based on the rules and information enclosed in the blueprint.

Class car: #class

Pass

Ford=car() #object

class Person: #**class**

# constructor

def \_\_init\_\_(self, name):

self.name = name

# Sample Method

def say\_hi(self):

print('Hello, my name is', self.name)

ahmed = Person('ahmed') #**object**

ahmed.say\_hi()