# CIT Python Week 6 Quiz

Total points 150/200

End of Week 6 Python Quiz

✗ Full Names * ···/5

Mbalire Shawal ✗

✗ 1. ____ represents an entity in the real world with its identity and behaviour. *0/5

- a) A method
- b) An object
- c) A class ✗
- d) An operator

Correct answer

- b) An object

Feedback

*An object represents an entity in the real world that can be distinctly identified. A class may define an object.*

✗ 2. ____ is used to create an object. * 0/5

- (●) a) class ✗
- ( ) b) constructor
- ( ) c) User-defined functions
- ( ) d) In-built functions

Correct answer

- (●) b) constructor

Feedback

*The values assigned by the constructor to the class members is used to create the object.*

✓ 3. `class Test:` * 5/5

```
    def __init__(self,a="Hello World"):
        self.a=a

    def display(self):
        print(self.a)
obj=Test()
obj.display()
```

- ○ a) The program has an error because constructor can't have default arguments
- ○ b) Nothing is displayed
- ◉ c) "Hello World" is displayed ✓
- ○ d) The program has an error display function doesn't have parameters

Feedback

*The program has no error. "Hello World" is displayed.*

✓ 4. What is setattr() used for? * 5/5

- ○ a) To access the attribute of the object
- ◉ b) To set an attribute ✓
- ○ c) To check if an attribute exists or not
- ○ d) To delete an attribute

Feedback

*setattr(obj,name,value) is used to set an attribute. If attribute doesn't exist, then it would be created.*

✓ 5. What is getattr() used for? * 5/5

- (●) a) To access the attribute of the object ✓
- ( ) b) To delete an attribute
- ( ) c) To check if an attribute exists or not
- ( ) d) To set an attribute

Feedback

*getattr(obj,name) is used to get the attribute of an object.*

✓ 6. What will be the output of the following Python code? * 5/5

```
class change:
    def __init__(self, x, y, z):
        self.a = x + y + z

x = change(1,2,3)
y = getattr(x, 'a')
setattr(x, 'a', y+1)
print(x.a)
```

- ( ) a) 6
- (●) b) 7 ✓
- ( ) c) Error
- ( ) d) 0

Feedback

*First, a=1+2+3=6. Then, after setattr() is invoked, x.a=6+1=7.*

✓ 7. What will be the output of the following Python code? * 5/5

```
class test:
    def __init__(self,a):
        self.a=a

    def display(self):
        print(self.a)
obj=test()
obj.display()
```

- a) Runs normally, doesn't display anything
- b) Displays 0, which is the automatic default value
- c) Error as one argument is required while creating the object ✓
- d) Error as display function requires additional argument

Feedback

*Since, the __init__ special method has another argument a other than self, during object creation, one argument is required. For example: obj=test("Hello")*

✓ 8. Is the following Python code correct? * 5/5

```
>>> class A:
        def __init__(self,b):
            self.b=b
        def display(self):
            print(self.b)
>>> obj=A("Hello")
>>> del obj
```

- (●) True ✓
- ( ) False

Feedback

*It is possible to delete an object of the class. On further typing obj in the python shell, it throws an error because the defined object has now been deleted.*

✓ 9. What will be the output of the following Python code? * 5/5

```
class test:
    def __init__(self):
        self.variable = 'Old'
        self.Change(self.variable)
    def Change(self, var):
        var = 'New'
obj=test()
print(obj.variable)
```

- a) Error because function change can't be called in the __init__ function
- b) 'New' is printed
- c) 'Old' is printed ✓
- d) Nothing is printed

Feedback

*This is because strings are immutable. Hence any change made isn't reflected in the original string.*

✓ 10. What is Instantiation in terms of OOP terminology? * 5/5

- a) Deleting an instance of class
- b) Modifying an instance of class
- c) Copying an instance of class
- d) Creating an instance of class ✓

Feedback

*Instantiation refers to creating an object/instance for a class.*

✓ 11. What will be the output of the following Python code? * 5/5

```
class fruits:
    def __init__(self, price):
        self.price = price
obj=fruits(50)

obj.quantity=10
obj.bags=2

print(obj.quantity+len(obj.__dict__))
```

- a) 12
- b) 52
- c) 13 ✓
- d) 60

Feedback

*In the above code, obj.quantity has been initialised to 10. There are a total of three items in the dictionary, price, quantity and bags. Hence, len(obj.__dict__) is 3.*

✓ 12. What will be the output of the following Python code? * 5/5

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

  def test(self):
    print(__name__)

obj = Demo()
obj.test()
```

- ○ a) Exception is thrown
- ◉ b) __main__ ✓
- ○ c) Demo
- ○ d) test

Feedback

*Since the above code is being run not as a result of an import from another module, the variable will have value "__main__".*

✓ 13. In which of the following does the CricketFan class correctly inherit from the PartyAnimal class? *5/5

- ○ A. from party import PartyAnimal
- ◉ B. class CricketFan(PartyAnimal) ✓
- ○ C. an = PartyAnimal()
- ○ D. CricketFan = PartyAnimal()

✓ 14. What will be the output of the following Python code? * 5/5

```
try:
    if '1' != 1:
        raise "someError"
    else:
        print("someError has not occurred")
except "someError":
    print ("someError has occurred")
```

- ○ a) someError has occurred
- ○ b) someError has not occurred
- ◉ c) invalid code ✓
- ○ d) none of the mentioned

Feedback

*A new exception class must inherit from a BaseException. There is no such inheritance here.*

✓ 15. What does the following code output? * 5/5

```
class People():
        def __init__(self, name):
            self.name = name

        def namePrint(self):
            print(self.name)

person1 = People("Sally")
person2 = People("Louise")
person1.namePrint()
```

- (●) A. Sally ✓
- ( ) B. Louise
- ( ) C. Sally Louise
- ( ) D. person1

✗ 16. Which of the following statements is **not** true about object-oriented programming? *0/5

- (●) A. One of the benefits of object-oriented programming is that it can hide complexity. ✗
- ( ) B. A class contains functions as well as the data that is used by those functions.
- ( ) C. Constructor methods are required to initialize an object and destructor methods are required to destroy the object when no longer required.
- ( ) D. A powerful feature of object-oriented programming is the ability to create a new class by extending an existing class.

Correct answer

- (●) C. Constructor methods are required to initialize an object and destructor methods are required to destroy the object when no longer required.

✓ 17. What is the output of the following code? * 5/5

```
class Pokemon():

    def __init__(self, name, type):
        self.name = name
        self.type = type

    def stringPokemon(self):
        print(f"Pokemon name is {self.name} and type is {self.type}")

class GrassType(Pokemon):

    # overrides the stringPokemon() function on 'Pokemon' class
    def stringPokemon(self):
        print(f"Grass type pokemon name is {self.name}")

poke1 = GrassType('Bulbasaur', 'Grass')
poke1.stringPokemon
poke1.stringPokemon()
poke2 = Pokemon('Charizard', 'Fire')
poke2.stringPokemon
poke2.stringPokemon()
```

- (●) A. Grass type pokemon name is Bulbasaur Pokemon name is Charizard and type is Fire ✓
- ( ) B. Pokemon name is Bulbasaur and type is Grass Pokemon name is Charizard and type is Fire
- ( ) C. Grass type pokemon name is Bulbasaur Grass type pokemon name is Charizard
- ( ) D. Error because the extending class has a stringPokemon() function which already exists.

✗ 18. True or False? In order to extend a class, the new class should have access to all the data and inner workings of the parent class. *0/5

- (●) True ✗
- ( ) False

Correct answer

- (●) False

✓ 19. Which of the following is the correct way to define an initializer method? *5/5

- ( ) A. def __init__(title, author):
- (●) B. def __init__(self, title, author): ✓
- ( ) C. def __init__():
- ( ) D. __init__(self, title, author):

✗ 20. Which of the following is correct with respect to OOP concept in Python? *0/5

- ( ) A. Objects are real world entities while classes are not real.
- ( ) B. Classes are real world entities while objects are not real.
- ( ) C. Both objects and classes are real world entities.
- (●) D. Both object and classes are not real. ✗

Correct answer

- (●) A. Objects are real world entities while classes are not real.

✓ 21. Who developed the Python language? * 5/5

- ○ Zim Den
- ◉ Guido van Rossum ✓
- ○ Niene Stom
- ○ Wick van Rossum

✓ 22. Which of the following is correct? * 5/5

```
class Book:
        def __init__(self,author):
                self.author=author
book1=Book("V.M.Shah")
book2=book1
```

- ○ A. Both book1 and book2 will have reference to two different objects of class Book.
- ◉ B. id(book1) and id(book2) will have same value. ✓
- ○ C. It will throw error as multiple references to same object is not possible.
- ○ D. None of the above

Feedback

*Ans : B*
*Explanation: book1 and book2 will reference to the same object. Hence, id(book1) and id(book2) will have same value.*

✗ 23. In python, what is method inside class? * 0/5

- (●) A. attribute ✗
- ( ) B. object
- ( ) C. argument
- ( ) D. function

Correct answer

- (●) D. function

✓ 24. What will be the output of below Python code? * 5/5

```
class A:
    def __init__(self,num):
        num=3
        self.num=num
    def change(self):
        self.num=7
a=A(5)
print(a.num)
a.change()
print(a.num)
```

- A. 5, 7
- B. 5, 5
- C. 3, 3
- D. 3, 7 ✓

✗ 25. Which one of the following syntaxes is the correct syntax to read from a simple text file stored in "d:\java.txt"? *0/5

- A. Infile = open("d:\\java.txt", "r")
- B. Infile = open(file="d:\\\java.txt", "r")
- C. Infile = open("d:\java.txt","r")
- D. Infile = open.file("d:\\java.txt","r") ✗

Correct answer

- A. Infile = open("d:\\java.txt", "r")

✓ 26. Study the following program: * 5/5

```
class book:
   def __init__(a, b):
      a.o1 = b

class child(book):
   def __init__(a, b):
      a.o2 = b

obj = page(32)
print "%d %d" % (obj.o1, obj.o2)
```

- 32
- 32 32
- 32 None
- Error is generated ✓

✗ 27. Study the following program: * 0/5

```
class Std_Name:
    def __init__(self, Std_firstName, Std_Phn, Std_lastName):
        self.Std_firstName = Std_firstName
        self.Std_PhnStd_Phn = Std_Phn
        self.Std_lastName = Std_lastName

Std_firstName = "Wick"
name = Std_Name(Std_firstName, 'F', "Bob")
Std_firstName = "Ann"
name.lastName = "Nick"
print(name.Std_firstName, name.Std_lastName)
```

What will be the output of this statement?

- ○ Ann Bob
- ○ Ann Nick
- ◉ Wick Bob ✗
- ○ Wick Nick

Correct answer

- ◉ Wick Nick

✓ 28. Study the following program: * 5/5

```
i = 1:
while True:
    if i%3 == 0:
        break
    print(i)
```

What will be the output of this statement?

- 1 2 3
- 3 2 1
- 1 2
- Invalid syntax ✓

✓ 29. Study the following program: * 5/5

```
i = 0
while i < 5:
    print(i)
    i += 1
    if i == 3:
        break
else:
    print(0)
```

What will be the output of this statement?

- 1 2 3
- 0 1 2 3
- 0 1 2 ✓
- 3 2 1

✗ 30. What error will occur when you execute the following code? * 0/5

MANGO = APPLE

- ○ NameError
- ○ SyntaxError
- ○ TypeError
- ◉ ValueError ✗

Correct answer

- ◉ NameError

✓ 31. Study the following statement * 5/5

z = {"x":0, "y":1}

Which of the following is the correct statement?

- ○ dictionary z is created
- ○ x and y are the keys of dictionary z
- ○ 0 and 1 are the values of dictionary z
- ◉ All of the above ✓

✓ 32. Which of the following is not a class method? * 5/5

- ◉ (a) Non-static ✓
- ○ (b) Static
- ○ (c) Bounded
- ○ (d) Unbounded

✓ 33. Which of the following best describes inheritance? * 5/5

- (a) Ability of a class to derive members of another class as a part of its own definition ✓
- (b) Means of bundling instance variables and methods in order to restrict access to certain class members
- (c) Focuses on variables and passing of variables to functions
- (d) Allows for implementation of elegant software that is well designed and easily modified

✓ 34. Which of the following statements isn't true? * 5/5

- (a) A non-private method in a superclass can be overridden
- (b) A derived class is a subset of superclass
- (c) The value of a private variable in the superclass can be changed in the subclass ✓
- (d) When invoking the constructor from a subclass, the constructor of superclass is automatically invoked

✓ 35. What will be the output of the following Python code? * 5/5

```
class A:
    def __init__(self):
        self.multiply(15)
        print(self.i)

    def multiply(self, i):
        self.i = 4 * i;
class B(A):
    def __init__(self):
        super().__init__()

    def multiply(self, i):
        self.i = 2 * i;
obj = B()
```

- (a) 15
- (b) 60
- (c) An exception is thrown
- (d) 30 ✓

✓ 36. What will be the output of the following Python code? * 5/5

```
class Demo:
    def check(self):
        return " Demo's check "
    def display(self):
        print(self.check())
class Demo_Derived(Demo):
    def check(self):
        return " Derived's check "
Demo().display()
Demo_Derived().display()
```

- (a) Demo’s check Derived’s check ✓
- (b) Demo’s check Demo’s check
- (c) Derived’s check Demo’s check
- (d) Syntax error

✓ 37. Which of the following is the most suitable definition for encapsulation? *5/5

- (a) Ability of a class to derive members of another class as a part of its own definition
- (b) Means of bundling instance variables and methods in order to restrict access to certain class members ✓
- (c) Focuses on variables and passing of variables to functions
- (d) Allows for implementation of elegant software that is well designed and easily modified

✓ 38. Methods of a class that provide access to private members of the class are called as _____ and _____ *5/5

- [x] (a) getters/setters
- [ ] (b) __repr__/__str__
- [ ] (c) user-defined functions/in-built functions
- [ ] (d) __init__/__del__

✓ 39. Private members of a class cannot be accessed. * 5/5

- [ ] True
- [x] False