

Python Training – Mid Assessment Test

Created by Ankur Saxena

Date – 01/02/2023

Question 1

What is an abstract class?

- ☐ An abstract class is the name for any class from which you can instantiate an object.
- ☐ Abstract classes must be redefined any time an object is instantiated from them.
- ☐ Abstract classes must inherit from concrete classes.
- ☐ An abstract class exists only so that other "concrete" classes can inherit from the abstract class.

Question 2

What happens when you use the build-in function `any()` on a list?

- ☐ The `any()` function will randomly return any item from the list.
- ☐ The `any()` function returns True if any item in the list evaluates to True. Otherwise, it returns False.
- ☐ The `any()` function takes as arguments the list to check inside, and the item to check for. If "any" of the items in the list match the item to check for, the function returns True.
- ☐ The `any()` function returns a Boolean value that answers the question "Are there any items in this list?"

Question 3

What statement about static methods is true?

What statement about static methods is true?

- ☐ Static methods are called static because they always return None.
- ☐ Static methods can be bound to either a class or an instance of a class.
- ☐ Static methods serve mostly as utility methods or helper methods, since they can't access or modify a class's state.
- ☐ Static methods can access and modify the state of a class or an instance of a class.

Question 4

What are attributes?

- ☐ Attributes are long-form version of an if/else statement, used when testing for equality between objects.
- ☐ Attributes are a way to hold data or describe a state for a class or an instance of a class.
- ☐ Attributes are strings that describe characteristics of a class.
- ☐ Function arguments are called "attributes" in the context of class methods and instance methods.

Question 5

What is the term to describe this code? `count, fruit, price = (2, 'apple', 3.5)`

☐ tuple assignment

☐ tuple unpacking

☐ tuple matching

☐ tuple duplication

Question 6

What built-in list method would you use to remove items from a list?

☐ `.delete()` method

☐ `pop(my_list)`

☐ `del(my_list)`

☐ `.pop()` method

Question 7

What is one of the most common use of Python's sys library?

- ☐ to capture command-line arguments given at a file's runtime
- ☐ to connect various systems, such as connecting a web front end, an API service, a database, and a mobile app
- ☐ to take a snapshot of all the packages and libraries in your virtual environment
- ☐ to scan the health of your Python ecosystem while inside a virtual environment

Question 8

What is the runtime of accessing a value in a dictionary by using its key?

- ☐ $O(n)$, also called linear time.
- ☐ $O(\log n)$, also called logarithmic time.
- ☐ $O(n^2)$, also called quadratic time.
- ☐ $O(1)$, also called constant time.

Question 9

What is the correct syntax for defining a class called Game, if it inherits from a parent class called LogicGame?

☐ `class Game(LogicGame): pass`

☐ `def Game(LogicGame): pass`

☐ `def Game.LogicGame(): pass`

☐ `class Game.LogicGame(): pass`

Question 10

What built-in Python data type is commonly used to represent a stack?

☐ `set`

☐ `list`

☐ `None`

☐ `dictionary`

Question 11

What would this expression return?

```
1 | college_years = ['Freshman', 'Sophomore', 'Junior', 'Senior']  
2 | return list(enumerate(college_years, 2019))
```

- ☐ [('Freshman', 2019), ('Sophomore', 2020), ('Junior', 2021), ('Senior', 2022)]
- ☐ [(2019, 2020, 2021, 2022), ('Freshman', 'Sophomore', 'Junior', 'Senior')]
- ☐ [('Freshman', 'Sophomore', 'Junior', 'Senior'), (2019, 2020, 2021, 2022)]
- ☐ [(2019, 'Freshman'), (2020, 'Sophomore'), (2021, 'Junior'), (2022, 'Senior')]

Question 12

What is the purpose of the "self" keyword when defining or calling instance methods?

- ☐ self means that no other arguments are required to be passed into the method.
- ☐ There is no real purpose for the self method; it's just historic computer science jargon that Python keeps to stay consistent with other programming languages.
- ☐ self refers to the instance whose method was called.
- ☐ self refers to the class that was inherited from to create the object using self.

Question 13

Which of these is NOT a characteristic of namedtuples?

- ☐ You can assign a name to each of the namedtuple members and refer to them that way, similarly to how you would access keys in dictionary.
- ☐ Each member of a namedtuple object can be indexed to directly, just like in a regular tuple.
- ☐ namedtuples are just as memory efficient as regular tuples.
- ☐ No import is needed to use namedtuples because they are available in the standard library.

Question 14

What is an instance method?

- ☐ Instance methods can modify the state of an instance or the state of its parent class.
- ☐ Instance methods hold data related to the instance.
- ☐ An instance method is any class method that doesn't take any arguments.
- ☐ An instance method is a regular function that belongs to a class, but it must return None.

Question 15

Which statement does NOT describe the object-oriented programming concept of encapsulation?

- ☐ It protects the data from outside interference.
- ☐ A parent class is encapsulated and no data from the parent class passes on to the child class.
- ☐ It keeps data and the methods that can manipulate that data in one place.
- ☐ It only allows the data to be changed by methods.

Question 16

What built-in Python data type is best suited for implementing a queue?

- ☐ dictionary
- ☐ set
- ☐ None. You can only build a queue from scratch.
- ☐ list

Question 17

What is the correct syntax for instantiating a new object of the type Game?

☐ `my_game = class.Game()`

☐ `my_game = class(Game)`

☐ `my_game = Game()`

☐ `my_game = Game.create()`

Question 18

What does the built-in map() function do?

☐ It creates a path from multiple values in an iterable to a single value.

☐ It applies a function to each item in an iterable and returns the value of that function.

☐ It converts a complex value type into simpler value types.

☐ It creates a mapping between two different elements of different iterables.

Question 19

If you don't explicitly return a value from a function, what happens?

☐ The function will return a RuntimeError if you don't return a value.

☐ If the return keyword is absent, the function will return None.

☐ If the return keyword is absent, the function will return True.

☐ The function will enter an infinite loop because it won't know when to stop executing its code.

Question 20

What is the purpose of the pass statement in Python?

- ☐ It is used to skip the yield statement of a generator and return a value of None.
- ☐ It is a null operation used mainly as a placeholder in functions, classes, etc.
- ☐ It is used to pass control from one statement block to another.
- ☐ It is used to skip the rest of a while or for loop and return to the start of the loop.

Question 21

What is the term used to describe items that may be passed into a function?

- ☐ arguments
- ☐ paradigms
- ☐ attributes
- ☐ decorators

Question 22

Implement a function called `maximum()` that returns the maximum of three numbers. Use conditional statement.

Example

```
[IN]: maximum(4, 2, 1)
```

```
[OUT]: 4
```

Question 23

Implement a function called `multi()`, which accepts an iterable object (list, tuple) as an argument and returns the product of all elements of this iterable object.

Example

1. [IN]: `multi((-4, 6, 2))`
2. [OUT]: `-48`

Question 24

Implement a function `map_longest()` that accepts the list of words and return the length of the longest word in this list.

1. [IN]: `map_longest(['python', 'sql'])`
2. [OUT]: `6`

Question 25

Implement a function called `filter_ge_6()` that takes a list of words and returns list of words with the length greater than or equal to 6 characters.

1. [IN]: `filter_ge_6(['programming', 'python', 'java', 'sql'])`
2. [OUT]: `['programming', 'python']`

1. [IN]: `filter_ge_6(['java', 'sql'])`
2. [OUT]: `[]`

Question 26

Implement a function called `factorial()` that calculates the factorial for a given number.

1. [IN]: `factorial(6)`
2. [OUT]: `720`

Question 27

Implement a function `count_str()`, which returns the number of *str* objects in an iterable object (list, tuple, set).

1. [IN]: `count_str(['p', 2, 4.3, None])`
2. [OUT]: `1`

Question 28

Implement a function `count_str()`, which returns the number of *str* objects with a length more than 2 characters from an iterable object (list, tuple, set).

1. [IN]: `count_str([1, '#hello', '', 'python', 'go'])`
2. [OUT]: 2

Question 29

The following list is given:

```
items = [(3, 4), (2, 5), (1, 4), (6, 1)]
```

Sort the list by the growing sum of squares of numbers in each tuple. Use the `sort()` method and the *lambda* expression and print sorted list to the console.

Expected result:

```
[(1, 4), (3, 4), (2, 5), (6, 1)]
```

Question 30

The present value - p_v and the investment period - n are given below:

1. $p_v = 1000$
2. $n = 10$

Depending on the interest rates given below, calculate the future value f_v of your investment:

1. $\text{rate} = [0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07]$

Round the result to the full cent and print the result to the console.

Expected result:

1. [1104.62, 1218.99, 1343.92, 1480.24, 1628.89, 1790.85, 1967.15]

