

---

# Python Programming Internal Paper

1. Which of the following statements is correct regarding the object-oriented programming concept in Python? \*

(1 Point)

- ☐ Classes are real-world entities while objects are not real
- ☒ Objects are real-world entities while classes are not real
- ☐ Both objects and classes are real-world entities
- ☐ All of the above

2. Which of the following operators is the correct option for power(ab)? \*

(1 Point)

- ☐  $a^b$
- ☒  $a^{**}b$
- ☐  $a^{^^}b$
- ☐  $a^{^*}b$

3. Study the following program:

```
z = "xyz"
j = "j"
while j in z:
    print(j, end=" ") *
```

(1 Point)

- ☐ xyz

☒ No output

☐ x y z

☐ j j j j j j j..

4. Study the following program:

```
i = {4, 5, 6}
```

```
i.update({2, 3, 4})
```

```
print(i)
```

What will be the output of this program? \*

(1 Point)

☐ 2 3 4 4 5 6

☒ 2 3 4 5 6

☐ 4 5 6 2 3 4

☐ Error, duplicate element presents in list

5. Study the following program:

```
mytuple1 = (2, 4, 3)
```

```
mytuple3 = mytuple1 * 2
```

```
print(mytuple3) *
```

(1 Point)

☒ (2, 4, 3, 2, 4, 3)

☐ (2, 2, 4, 4, 3, 3)

☐ (4, 8, 6)

☐ Error

6. What will be the output of the following Python code?

```
def mk(x):  
    def mk1():  
        print("Decorated")  
        x()  
    return mk1  
def mk2():  
    print("Ordinary")  
p = mk(mk2)  
p() *
```

(1 Point)

- ☐ Decorated Decorated
- ☐ Ordinary Decorated
- ☐ Ordinary Ordinary
- ☒ Decorated Ordinary

7. The \_\_\_\_\_ symbol along with the name of the decorator function can be placed above the definition of the function to be decorated works as an alternate way for decorating a function. \*

(1 Point)

- ☐ a) #
- ☐ b) \$
- ☒ c) @
- ☐ d) &

8. The following python code can work with \_\_\_\_\_ parameters.

```
def f(x):  
    def f1(*args, **kwargs):  
        print("foundry")  
        return x(*args, **kwargs)  
    return f1 *
```

(1 Point)

- ☐ 2
- ☐ 1
- ☒ any number of
- ☐ 0

9. What will be the output of the following Python code?

```
class Count:
    def __init__(self, count = 0):
        self.__count = count
```

```
c1 = Count(2)
c2 = Count(2)
print(id(c1) == id(c2), end = " ")
```

```
s1 = "Good"
s2 = "Good"
print(id(s1) == id(s2)) *
```

(1 Point)

- ☐ a) True False
- ☐ b) True True
- ☒ c) False True
- ☐ d) False False

10. What will be the output of the following Python code?

```
print("xyyzxyzxxyy".count('yy')) *
```

(1 Point)

- ☒ a) 2
- ☐ b) 0
- ☐ c) error
- ☐ d) none of the mentioned

11. What will be the output of the following Python code?

```
print("Hello {1} and {0}".format('bin', 'foo')) *
```

(1 Point)

- ☒ a) Hello foo and bin
- ☐ b) Hello bin and foo
- ☐ c) Error
- ☐ d) None of the mentioned

12. Suppose list1 is [2, 33, 222, 14, 25], What is list1[-1]? \*

(1 Point)

- ☐ a) Error
- ☐ b) None
- ☒ c) 25
- ☐ d) 2

13. Suppose listExample is [3, 4, 5, 20, 5, 25, 1, 3], what is list1 after listExample.pop()? \*

(1 Point)

- ☒ a) [3, 4, 5, 20, 5, 25, 1]
- ☐ b) [1, 3, 3, 4, 5, 5, 20, 25]
- ☐ c) [3, 5, 20, 5, 25, 1, 3]
- ☐ d) [1, 3, 4, 5, 20, 5, 25]

14. What will be the output of the following Python code?

```
veggies = ['carrot', 'broccoli', 'potato', 'asparagus']  
veggies.insert(veggies.index('broccoli'), 'celery')  
print(veggies) *
```

(1 Point)

- ☒ a) ['carrot', 'celery', 'broccoli', 'potato', 'asparagus']
- ☐ b) ['carrot', 'celery', 'potato', 'asparagus']
- ☐ c) ['carrot', 'broccoli', 'celery', 'potato', 'asparagus']
- ☐ d) ['celery', 'carrot', 'broccoli', 'potato', 'asparagus']

15. Write a list comprehension for number and its cube for l=[1, 2, 3, 4, 5, 6, 7, 8, 9]. \*

(1 Point)

- ☒ a) [x\*\*3 for x in l]
- ☐ b) [x^3 for x in l]
- ☐ c) [x\*\*3 in l]
- ☐ d) [x^3 in l]

16. What will be the output of the following Python code?

```
>>> a,b=6,7
>>> a,b=b,a
>>> a,b *
```

(1 Point)

- ☐ a) (6,7)
- ☐ b) Invalid syntax
- ☒ c) (7,6)
- ☐ d) Nothing is printed

17. If a=(1,2,3,4), a[1:-1] is \_\_\_\_\_ \*

(1 Point)

- ☒ a) Error, tuple slicing doesn't exist
- ☐ b) [2,3]
- ☐ c) (2,3,4)

☐ d) (2,3)

18. What will be the output of the following Python code?

```
s1={1, 2, 3}
s2={3, 4, 5, 6}
s1.difference(s2)
s2.difference(s1) *
```

(1 Point)

- ☒ a) {1, 2} {4, 5, 6}
- ☐ b) {1, 2} {1, 2}
- ☐ c) {4, 5, 6} {1, 2}
- ☐ d) {4, 5, 6} {4, 5, 6}

19. How many keyword arguments can be passed to a function in a single function call? \*

(1 Point)

- ☐ a) zero
- ☐ b) one
- ☒ c) zero or more
- ☐ d) one or more

20. How are variable length arguments specified in the function heading? \*

(1 Point)

- ☒ a) one star followed by a valid identifier
- ☐ b) one underscore followed by a valid identifier
- ☐ c) two stars followed by a valid identifier
- ☐ d) two underscores followed by a valid identifier

21. Which function overloads the + operator? \*

(1 Point)

- ☒ a) `__add__()`
- ☐ b) `__plus__()`
- ☐ c) `__sum__()`
- ☐ d) none of the mentioned

22. What is the difference between r+ and w+ modes? \*

(1 Point)

- ☒ a) no difference
- ☐ b) in r+ the pointer is initially placed at the beginning of the file and the pointer is at the end for w+
- ☐ c) in w+ the pointer is initially placed at the beginning of the file and the pointer is at the end for r+
- ☐ d) depends on the operating system

23. How do you get the current position within the file? \*

(1 Point)

- ☐ a) `fp.seek()`
- ☒ b) `fp.tell()`
- ☐ c) `fp.loc`
- ☐ d) `fp.pos`

24. 9. How do you change the file position to an offset value from the start? \*

(1 Point)

- ☐ a) `fp.seek(offset, 0)`
- ☐ b) `fp.seek(offset, 1)`



- ☐ c) [fp.seek](#)(offset, 2)
- ☒ d) none of the mentioned

25. What will be the output of the following Python code?

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

(1 Point)

- ☐ 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

26. 8. Suppose B is a subclass of A, to invoke the \_\_init\_\_ method in A from B, what is the line of code you should write? \*

(1 Point)

- ☒ a) A.\_\_init\_\_(self)
- ☐ b) B.\_\_init\_\_(self)
- ☐ c) A.\_\_init\_\_(B)
- ☐ d) B.\_\_init\_\_(A)

27. What will be the output of the following Python code?

```
class Test:
    def __init__(self):
        self.x = 0
class Derived_Test(Test):
    def __init__(self):
        Test.__init__(self)
        self.y = 1
def main():
    b = Derived_Test()
    print(b.x,b.y)
main() *
```

(1 Point)

- ☐ a) Error because class B inherits A but variable x isn't inherited
- ☐ b) 0 0
- ☐ c) 0 1
- ☒ d) Error, the syntax of the invoking method is wrong

28. What will be the output of the following Python code?

```
>>> class A:
    pass
>>> class B(A):
    pass
>>> obj=B()
>>> isinstance(obj,A) *
```

(1 Point)

- ☐ a) True
- ☐ b) False
- ☒ c) Wrong syntax for isinstance() method
- ☐ d) Invalid method for classes

29. What will be the output of the following Python code?

```
class Demo:
    def __init__(self):
        self.x = 1
    def change(self):
        self.x = 10
class Demo_derived(Demo):
    def change(self):
        self.x=self.x+1
        return self.x
def main():
    obj = Demo_derived()
    print(obj.change())
```

main() \*

(1 Point)

- ☐ a) 11
- ☒ b) 2
- ☐ c) 1
- ☐ d) An exception is thrown

30. What will be the output of the following Python code?

```
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() \*

(1 Point)

- ☐ a) 15
- ☒ b) 60
- ☐ c) An exception is thrown
- ☐ d) 30

31. What will be the output of the following Python code?

```
class A:
    def __init__(self, x, y):
        self.x = x
        self.y = y
    def __str__(self):
        return 1
    def __eq__(self, other):
        return self.x * self.y == other.x * other.y
obj1 = A(5, 2)
obj2 = A(2, 5)
print(obj1 == obj2) *
```

(1 Point)

- ☐ a) False
- ☐ b) 1
- ☒ c) True
- ☐ d) An exception is thrown

32. What will be the output of the following Python code?

```
class Demo:
    def __init__(self):
        self.a = 1
        self.__b = 1

    def get(self):
        return self.__b

obj = Demo()
print(obj.get()) *
```

(1 Point)

- ☐ a) The program has an error because there isn't any function to return self.a
- ☐ b) The program has an error because b is private and display(self) is returning a private member
- ☐ c) The program has an error because b is private and hence can't be printed
- ☒ d) The program runs fine and 1 is printed

33. What type of inheritance is illustrated in the following piece of code?

```
class A():  
    pass  
class B(A):  
    pass  
class C(B):  
    pass *
```

(1 Point)

- ☒ Multi-level inheritance
- ☐ Multiple inheritance
- ☐ Hierarchical inheritance
- ☐ Single-level inheritance

34. Which of the following is the most suitable definition for encapsulation? \*

(1 Point)

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

35. Methods of a class that provide access to private members of the class are called as \_\_\_\_\_ and \_\_\_\_\_ \*

(1 Point)

- ☒ getters/setters

- ☐ \_\_repr\_\_ / \_\_str\_\_
- ☐ user-defined functions/in-built functions
- ☐ \_\_init\_\_ / \_\_del\_\_

---

This content is created by the owner of the form. The data you submit will be sent to the form owner. Microsoft is not responsible for the privacy or security practices of its customers, including those of this form owner. Never give out your password.

Powered by Microsoft Forms |

The owner of this form has not provided a privacy statement as to how they will use your response data. Do not provide personal or sensitive information.

| [Terms of use](#)