

## Ôn Tập

### Ngôn ngữ lập trình Python

#### Câu 1: Khẳng định nào sau đây là đúng về Python ?

- Python là một ngôn ngữ thông dịch cấp cao
- Python là ngôn ngữ lập trình hướng đối tượng
- Python là ngôn ngữ mã nguồn mở
- Tất cả các đáp án đều đúng

#### Câu 2: Khẳng định nào sau đây là không đúng về Python ?

- Python có thể sử dụng trên nhiều hệ điều hành khác nhau: Unix, Windows, Mac OS, Linux, ...
- Python là một ngôn ngữ không phân biệt kiểu chữ HOA, chữ thường
- Python hoàn toàn tạo kiểu động và dùng cơ chế cấp phát bộ nhớ tự động
- Python có tốc độ thực hiện chậm hơn nhiều lần so với các ngôn ngữ biên dịch như C, Java, ...

#### Câu 3: Cho biết kết quả của đoạn code sau

```
for i in range(1,5):  
    print(i,end=' ')  
    if i == 3:  
        break
```

- IndentationError: expected an indented block
- 1 2 3
- 1 2 3 4
- 1 2

#### Câu 4: Các khối code (khối lệnh của hàm, vòng lặp,...) trong Python được xác định?

- Dấu ngoặc nhọn { }
- Canh lề
- Dấu ngoặc đơn ( )
- Dấu ngoặc vuông [ ]

#### Câu 5: Khẳng định nào là đúng về chú thích trong Python?

- Python sử dụng ký tự # để bắt đầu một chú thích
- Nội dung của chú thích sẽ được trình thông dịch bỏ qua
- Python dùng “”” ””” (3 cặp nháy đôi) hoặc """ """ (3 cặp nháy đơn) để viết chú thích trên nhiều dòng
- Tất cả các đáp án trên.

**Câu 6: Khẳng định nào là đúng về chú thích trong Python?**

- Python sử dụng ký tự // để bắt đầu một chú thích
- Python sử dụng ký tự % để bắt đầu một chú thích
- Python sử dụng ký tự # để bắt đầu một chú thích
- Python dùng “”” (2 cặp nháy đôi) hoặc “ ” (2 cặp nháy đơn) để viết chú thích trên nhiều dòng

**Câu 7: Kết quả của hàm sau đây là gì?**

```
print(type(type(float)))
```

- <class 'type'>
- <class 'float'>
- Có xuất hiện lỗi Error
- None

**Câu 8: Phép toán nào có thể được dùng để so sánh hai biến ?**

- ==
- \*\*
- //
- =

**Câu 9: Câu lệnh nào sau đây in ra kiểu dữ liệu của biến x?**

- print(type(x))
- print(type[x])
- print(typeof[x])
- print(typeof(x))

**Câu 10: Câu lệnh nào sau đây được dùng để gán giá trị cho x = 1, y = 2, z = 3 ?**

- x,y,z = 1;2;3
- x;y;z = 1;2;3
- x,y,z = 1,2,3
- x;y;z = 1,2,3

**Câu 11: Kết quả của câu lệnh sau là gì?**

```
print(0b1010)
```

- 10
- 0b1010
- 1010
- '0b1010'

**Câu 12: Kết quả của câu lệnh sau là gì?**

```
print(type(hex(15)))
```

- <class 'int'>
- <class 'str'>
- 0xf
- '0xf'

**Câu 13: Kết quả của câu lệnh sau là gì?**

```
print(bin(4))
```

- 100
- 0b100
- 0o100
- 0x100

**Câu 14: Kết quả của câu lệnh sau là gì? (Lưu ý: số lượng khoảng trắng trước số đó là 5)**

```
print(float('  -50\n'))
```

- -50
- -50.0
- Có xuất hiện lỗi Error
- -50.0000000000000000 (16 chữ số sau dấu chấm động)

**Câu 15: Kết quả của câu lệnh sau là gì?**

```
print(float('7e+002'))  
print(float('7e-002'))
```

- 700.0
- 0.07
- 700
- 0.07
- '7e+002'
- '7e-002'
- 7e+002

**Câu 16: Kết quả của câu lệnh sau là gì?**

```
print(complex('2+8j'))
```

- Có xuất hiện lỗi Error
- `complex('2+8j')`
- `'2+8j'`
- `(2+8j)`

**Câu 17: Chỉ ra trường hợp không hợp lệ khi đặt tên biến trong python ?**

- `bien-X = 5`
- `_bienX = 5`
- `bien_X = 5`
- `bienX = 5`

**Câu 18: Quy tắc nào sau đây là đúng khi đặt tên cho biến trong Python?**

- Tên biến có thể bắt đầu bằng một chữ số
- Tên biến có thể chứa dấu gạch dưới `"_"`
- Tên biến có thể có các ký hiệu đặc biệt như `!, @, #, $, %, ...`
- Tên biến có thể trùng với các từ khóa (keyword)

**Câu 19: Biến L trong đoạn sau là kiểu dữ liệu nào?**

```
L = "[5:2,10:[1,2]]"
```

- List (danh sách)
- String (chuỗi ký tự)
- Tuple
- Dictionary (từ điển)

**Câu 20: Kết quả của lệnh sau là:**

```
print(100, 200, "hello", "world", sep=':')
```

- `100:200:hello:world`
- `100200helloworld`
- `100 200 hello world`
- Có xuất hiện lỗi Error

**Câu 21: Lệnh nào dùng để nhập dữ liệu từ bàn phím trong Python?**

- cin
- scanf()
- **input()**
- read()

**Câu 22: Kết quả của lệnh sau là:**

```
print(100 >= 10**2)
```

- 100 >= 10\*\*2
- **True**
- False
- None

**Câu 23: Kết quả của lệnh sau là:**

```
print(10//3,10%3,10/3,sep=" , ")
```

- **3 , 1 , 3.3333333333333335**
- 1 , 3 , 3.3333333333333335
- 3.3333333333333335 , 1 , 3
- 3 , 3.3333333333333335 , 1

**Câu 24: Đây là output của chương trình dưới đây?**

```
def Function_1():
    global n
    n = 200
    def Function_2():
        global n
        n = 300
    n = 400

n = 500
Function_1()
print('n =', n)
```

- **n = 400**
- n = 500
- n = 200
- n = 300

**Câu 25: Cho biết kết quả của đoạn chương trình sau**

```
for i in range(1,5):  
    print(i, end=' ')  
    if i == 3:  
        break
```

- IndentationError: expected an indented block
- 1 2 3
- 1 2 3 4
- 1 2

**Câu 26: Cho biết kết quả đoạn chương trình sau**

```
if 'toan' in {'cntt': 1, 'toan': 2, 'sinh': 3}:  
    print(1)  
    if 'a' in 'cntt':  
        print(2)  
print(3)
```

- 1  
2  
3
- 1  
3
- 1  
2
- 3

**Câu 27: Cho biết output đoạn chương trình sau:**

```
i = 0  
x = 0  
while i < 5:  
    if i % 2 == 0:  
        x += 1  
    i += 1  
print(x)
```

- 3
- 10

- 5
- 0

**Câu 28: Kết quả nào là output của đoạn code dưới đây?**

```
for i in range(20):
    if i == 10:
        break
    else:
        print(i,end="")
else:
    print("*")
```

- 0123456789\*
- 123456789
- 0123456789
- 123456789\*

**Câu 29: Kết quả nào là output của đoạn code dưới đây?**

```
for i in range(5, 7):
    print(str(i) * 3)
```

- 555  
666
- 15  
18  
21
- 15  
18
- 555  
666  
777

**Câu 30: Điền phần còn thiếu trong đoạn code để được out dưới đây:**

```
#output:
66
44
```

```
for i in range(6, 0, ____):
    print(str(i) * 2)
```

- 0
- None
- -2
- -1

**Câu 31: Đây là kết quả của đoạn code dưới đây?**

```
myList = [1, 2, 3, 4, 5, 5, 1]
_max = myList[0]
index_max = 0
for i in range(1, len(myList)):
    if myList[i] > _max:
        _max = myList[i]
        index_max = i
print(index_max)
```

- 0
- 4
- 1
- 5

**Câu 32: Đây là kết quả của đoạn code dưới đây?**

```
myList = [1, 6, 3, 4, 5, 5, 1]
_max = myList[0]
index_max = 0
for i in range(1, len(myList)):
    if myList[i] > _max:
        _max = myList[i]
        index_max = i
print(index_max)
```

- 0
- 4
- 1
- 5

**Câu 33: Đây là kết quả của đoạn code dưới đây?**

```
myList = [1, 6, 3, 4, 5, 5, 1]
_max = myList[0]
index_max = 0
for i in range(2, len(myList)):
    if myList[i] > _max:
        _max = myList[i]
        index_max = i
print(index_max)
```



- 0
- 4
- 1
- 5

**Câu 34: Kết quả nào là output của đoạn code dưới đây?**

```
x = True
y = False
z = False
if not x or y:
    print (1)
elif not x or not y and z:
    print (2)
elif not x or y or not y and x:
    print (3)
else:
    print (4)
```

- 2
- 4
- 1
- 3

**Câu 35: Kết quả nào là output của đoạn code dưới đây?**

```
letter = ["a", "b"]
for i in enumerate(letter):
    print(i)
```

- (0, 'a')
- (1, 'b')
- a
- b
- 'a'
- 'b'
- 0
- 1

**Câu 36: Kết quả nào là output của đoạn code dưới đây?**

```
letter = ["a", "b"]
for i, j in enumerate(letter):
    print(i, j)
```

- (0, 'a')
- (1, 'b')

- a 0  
b 1
- ('a', 0)  
('b',1)
- 0 a  
1 b

**Câu 37: Kết quả nào là output của đoạn code dưới đây?**

```
for item in "HelloWorld":
    if item == "l":
        continue
    print(item,end="")
```

- HelloWorld
- HeloWorld
- HeoWord
- HeoWorld

**Câu 38: Kết quả nào là output của đoạn code dưới đây?**

```
for i in range(1,10):
    if (i%2!=0):
        continue
    print(i, end=" ")
```

- 2 4 6 8
- 1 3 5 7
- 1 3 5 7 9
- 2 4 6 8 10

**Câu 39: Kết quả nào là output của đoạn code dưới đây?**

```
word ='Python'
word[0] = 'K'
print(word[0])
```

- P
- K
- None
- Có xuất hiện lỗi Error

**Cau 40: Trong Python, hàm nào được dùng để thay thế một phần của chuỗi?**

- `replace()`
- `switch()`
- `split()`
- `replaceString()`

**Cau 41: Trong Python, hàm nào được dùng để chuyển đổi các ký tự in thường sang in hoa ?**

- `upper()`
- `uppercase()`
- `upperCase()`
- `toUpperCase()`

**Câu 42 : Chạy code sau trong Python, kết quả là:**

```
num = "1" * "5"  
print(num)
```

- 5
- 11111
- 55555
- `TypeError: can't multiply sequence by non-int of type 'str'`

**Câu 43: Chạy code sau trong Python, kết quả là:**

```
num = "1" + "5"  
print(num)
```

- 15
- 6
- 5
- `TypeError: can't multiply sequence by non-int of type 'str'`

**Câu 44 : Chạy code sau trong Python, kết quả là:**

```
num = "1" * 5  
print(num)
```

- 55555
- 11111
- 5
- `TypeError: can't multiply sequence by non-int of type 'str'`

**Câu 45: Output của lệnh dưới đây là:**

```
print("Hello World"[::-1])
```

- dlroW olleH
- Hello Worl
- d
- Có xuất hiện lỗi Error

**Câu 46: Kết quả của chương trình được in ra màn hình là?**

```
string = "my name is x"  
for i in string:  
    print (i, end=" ", )
```

- m, y, , n, a, m, e, , i, s, , x,
- m y, , n, a, m, e, , i, s, , x
- my, name, is, x,
- Error

**Cau 47: Trong Python, hàm nào được dùng để kết hợp các chuỗi ?**

- join()
- split()
- replace()
- strip()

**Câu 48: Kết quả của chương trình được in ra màn hình là?**

```
string="12345"  
print('*'.join(reversed(string)))
```

- 5\*4\*3\*2\*1
- 12345
- 54321
- 1\*2\*3\*4\*5

**Câu 49: Kết quả của chương trình được in ra màn hình là?**

```
word="hello world"
print(word.split('o'))
```

- ['hello w', 'rld' ]
- ['hell', ' w', 'rld']
- ['hello', 'world' ]
- ['hell', 'world' ]

**Câu 50: Kết quả của chương trình được in ra màn hình là?**

```
ss = "{1}, {0} va {2}".format('Hello','Hi','World')
print(ss)
```

- Hi, World va Hello
- Hello, World va Hi
- Hi, Hello va World
- Hello, Hi va World

**Câu 51: Kết quả của đoạn chương trình sau**

```
L = ['a','b','c','d','.', 'e']
print ("".join(L))
```

- abcde
- ['a','b','c','d','e']
- abcd.e
- a.b.c.d.e

**Câu 52: Kiểu dữ liệu nào sau đây là LIST?**

- ("apple", "banana", "cherry")
- {"apple", "banana", "cherry"}
- {"name": "apple", "color": "green"}

- ["apple", "banana", "cherry"]

**Câu 53: Kiểu dữ liệu nào sau đây là TUPLE?**

- ("apple", "banana", "cherry")
- {"apple", "banana", "cherry"}
- {"name": "apple", "color": "green"}
- ["apple", "banana", "cherry"]

**Câu 54: Kiểu dữ liệu nào sau đây là SET?**

- ("apple", "banana", "cherry")
- {"apple", "banana", "cherry"}
- {"name": "apple", "color": "green"}
- ["apple", "banana", "cherry"]

**Câu 55: Kiểu dữ liệu nào sau đây là DICTIONARY?**

- ("apple", "banana", "cherry")
- {"apple", "banana", "cherry"}
- {"name": "apple", "color": "green"}
- ["apple", "banana", "cherry"]

**Câu 56: Kiểu dữ liệu nào sau đây chứa các phần tử có thứ tự, có thể được thay đổi và cho phép trùng nhau**

- LIST
- DICTIONARY
- TUPLE
- SET

**Câu 57: Kết quả của hàm hiển thị dưới đây là gì?**

```
print(len(["hello", 2, 4, 6]))
```

- 4
- 3
- 8
- None

**Câu 58: Giả sử có một list: l = ["hello", 2, 4, 6]. Nếu muốn in list này theo thứ tự ngược lại ta nên sử dụng phương pháp nào sau đây?**

- reverse(l)
- list(reverse(l))
- reversed(l)

- `list(reversed(l))`

**Câu 59: Kết quả của hàm hiển thị dưới đây là gì?**

```
print(list(enumerate([2, 3])))
```

- `[2, 3]`
- `[(1, 2), (2, 3)]`
- `[(0, 2), (1, 3)]`
- `[(2, 3)]`

**Câu 60: Đâu là điểm khác biệt giữa tuple và list?**

- Tuple nằm trong dấu ngoặc nhọn `{ }`, list nằm trong dấu ngoặc vuông `[ ]`.
- Tuple nằm trong dấu ngoặc vuông `[ ]`, list nằm trong dấu ngoặc nhọn `{ }`.
- Tuple chứa các phần tử có thứ tự, list chứa các phần tử không có thứ tự.
- Dữ liệu thuộc kiểu list có thể thay đổi được, dữ liệu thuộc kiểu tuple không thể thay đổi được.

**Câu 61: Đâu là output của đoạn code dưới đây?**

```
mylist=[1, 5, 9, int('10')]  
print(sum(mylist))
```

- 25
- 15
- 10
- Có xuất hiện lỗi Error

**Câu 62: Kết quả của chương trình được in ra là:**

```
mylist=['a', 'aa', 'aaa', 'b', 'bb', 'bbb']  
print(mylist[:-1])
```

- `[a, aa, aaa, b, bb]`
- `['a', 'aa', 'aaa', 'b', 'bb', 'bbb']`
- `['a', 'aa', 'aaa', 'b', 'bb']`
- `[a, aa, aaa, b, bb, bbb]`

**Câu 63: Lệnh `print()` in ra kết quả nào cho chương trình dưới đây?**

```
list1 = [1, 3]
list2 = list1
list1[0] = 4
print(list2)
```

- [4, 3]
- [1, 3]
- [1, 4]
- [1, 3, 4]

**Câu 64: Đâu là giá trị của colors[2]?**

```
colors = ['red', 'orange', 'yellow', 'green', 'blue', 'indigo', 'violet']
```

- orange
- indigo
- blue
- yellow

**Câu 65 : Đâu không phải là kiểu dữ liệu tiêu chuẩn trong Python?**

- List
- Dictionary
- Class
- Tuple

**Câu 66: Kết quả nào là output của đoạn code dưới đây?**

```
list = [ 'Python', 100, 13.03, 'C++', 33.3 ]
print (list[1:3])
```

- ['100, 13.03, 'C++']
- [100, 13.03]
- ['Python', 'C++']
- ['Python', 100, 13.03]

**Câu 67: Đối tượng dưới đây thuộc kiểu dữ liệu nào?**

```
L = [1, 23, 'hello', 1]
```

- List
- Dictionary
- Tuple
- Set

**Câu 68 : Kết quả của đoạn code dưới đây là**



```
numbers = [2, 3, 4]
print(numbers)
```

- 2, 3, 4
- 2 3 4
- [2, 3, 4]
- [2 3 4]

**Câu 69: Theo dõi đoạn code dưới đây và chọn đáp án đúng nhất**

```
a = [1, 4, 20, 2, 5]
x = a[0]
for i in a:
    if i > x:
        x = i
print(x)
```

- x là giá trị trung bình của list.
- x là giá trị nhỏ nhất của list.
- x là giá trị lớn nhất của list.
- x là tổng giá trị các số trong list.

**Câu 70: Theo dõi đoạn code dưới đây và chọn đáp án đúng nhất**

```
s = set([2,2,2,1,1,3,4,5,5])
print(s)
```

- [2,2,2,1,1,3,4,5,5]
- {1, 2, 3, 4, 5}
- [2,1,3,4,5]
- {2,2,2,1,1,3,4,5,5}

**Câu 71: Theo dõi đoạn code dưới đây và chọn đáp án đúng nhất**

```
print({1,2,3,2,4,4,5} - {1,1,2,3}, {1,2,3,2,4,4,5,6,6}.difference({1,1,2,3}))
```

- {4, 5} {4, 5, 6}
- {1, 2, 3, 4, 5} {1, 2, 3, 4, 5, 6}
- {4, 5} {1, 2, 3}
- {1, 2, 3} {4, 5, 6}

**Câu 72: Theo dõi đoạn code dưới đây và chọn đáp án đúng nhất**

```
t = (1, 2, 3)
t[0] = 42
print(t[0])
```

- 1
- 42
- Có xuất hiện lỗi Error
- 0

**Câu 73: Theo dõi đoạn code dưới đây và chọn đáp án đúng nhất**

```
T = (1,2,3,4,5)
del T[3]
print(T)
```

- (1,2,3,4,5)
- (1,2,4,5)
- Có xuất hiện lỗi Error
- (1,2,3,5)

**Câu 74: Theo dõi đoạn code dưới đây và chọn đáp án đúng nhất**

```
d = dict()
d[2] = 100
d[2] = 200
d[2] = 400
d[0] = 100
d[1] = 200
d[0] = 300
print(d)
```

- {2: 700, 0: 400, 1: 200}
- {400:2, 300:0, 200:1}
- {2: 400, 0: 300, 1: 200}
- Có xuất hiện lỗi Error

**Câu 75: Theo dõi đoạn code dưới đây và chọn đáp án đúng nhất**

```
d = { 1:"a", 2:"b" }
print(d.get(0), d.get(0, 42))
```

- None 42
- Có xuất hiện lỗi Error
- None a

- None None

**Câu 76: Output của chương trình dưới đây là gì?**

```
def Find_max(a, b):  
    if a > b:  
        print(a, 'is maximum')  
    elif a == b:  
        print(a, 'is equal to', b)  
    else:  
        print(b, 'is maximum')  
Find_max(30.3, 40)
```

- 30.3 is maximum
- None
- 40 is maximum
- 30.3 is equal to 30.3

**Câu 77: Output của chương trình dưới đây là gì?**

```
def say(message, times = 1):  
    print(message * times)  
say('Hello')  
say('World', 5)
```

- Hello  
WorldWorldWorldWorldWorld
- Hello  
World 5
- Hello  
World,World,World,World,World
- Hello  
HelloHelloHelloHelloHello

**Câu 78: Output của chương trình dưới đây là gì?**

```
def func(a, b=5, c=10):  
    print('a =', a, 'và b =', b, 'và c =', c)  
func(3, 7)  
func(25, c = 24)  
func(c = 50, a = 100)
```

- a = 7 và b = 3 và c = 10  
a = 25 và b = 5 và c = 24  
a = 5 và b = 100 và c = 50
- a = 3 và b = 7 và c = 10  
a = 5 và b = 25 và c = 24  
a = 50 và b = 100 và c = 5
- a = 3 và b = 7 và c = 10  
a = 25 và b = 5 và c = 24  
a = 100 và b = 5 và c = 50
- a = 7 và b = 3 và c = 10  
a = 25 và b = 5 và c = 24  
a = 100 và b = 5 và c = 50

**Câu 79: Output của chương trình dưới đây là gì?**

```
def maximum(x, y):  
    if x > y:  
        return x  
    elif x == y:  
        return 'Các số bằng nhau'  
    else:  
        return y  
print(maximum(20, 30.3))
```

- 20
- 30.3
- Các số bằng nhau
- None

**Câu 80: Đây là lợi thế của việc sử dụng hàm trong Python?**

- Tránh việc phải lặp lại code thực thi những tác vụ tương tự nhau.
- Phân tách các vấn đề phức tạp thành các phần đơn giản hơn.
- Code rõ ràng, dễ quản lý hơn
- Tất cả các đáp án đều đúng

**Câu 81: Hàm có thể được khai báo ở đâu?**

- Trong module
- Trong Class
- Trong một hàm khác
- Tất cả các phương án trên

**Câu 82: Output của chương trình dưới đây là gì?**

```
def say(s1 = 'Hello World!', s2):  
    print(s1,s2)  
say(s2 = 'Hi', s1 = 'Hi World')  
say(s2 ="Hello Python")
```

- Hi World Hi  
Hello World! Hello Python
- Hi Hi World  
Hello Python Hello World!
- Hi World Hi  
Hello Python Hello World!
- Có xuất hiện lỗi Error

**Câu 83: Từ khóa nào được sử dụng để bắt đầu hàm?**

- fun
- define
- def
- function

**Câu 84: Chọn đáp án đúng: Phát biểu nào chính xác khi nói về Hàm trong Python?**

- Hàm có thể được tái sử dụng trong chương trình.
- Sử dụng hàm không có tác động tích cực gì đến các module trong chương trình.
- Không thể tự tạo các hàm của riêng người viết chương trình.
- Tất cả các đáp án trên đều đúng.

**Câu 85: Hàm nào sau đây là hàm tích hợp sẵn trong Python**

- cos()
- sqrt()
- factorial()
- print()

**Câu 86: Kết quả của câu lệnh sau là gì?**

```
print(round(4.567))
```

- 4.5
- 5
- 4
- 4.6

**Câu 87: Output của hàm biểu diễn dưới đây là gì?**

```
import math  
print(abs(math.sqrt(81)))
```

- None
- -9
- 9
- 9.0

**Câu 88: Mở file với chế độ mode 'a' có ý nghĩa gì?**

- Mở file ở chế độ chỉ được phép đọc.
- Mở file ở chế độ ghi.
- Mở file chế độ ghi tiếp vào cuối file.
- Mở file để đọc và ghi.

**Câu 89: Đoạn code dưới đây có ý nghĩa gì?**

```
f = open("test.txt")
```

- Mở file test.txt được phép đọc và ghi vào file.
- Mở file test.txt và chỉ được phép đọc file.
- Mở file test.txt và được phép ghi đè vào file
- Mở file test.txt và được phép ghi tiếp vào file.

**Câu 90: Mở file với chế độ mode 'wb' có ý nghĩa gì?**

- Mở file để ghi.
- Mở file để đọc và ghi.
- Mở file để ghi cho dạng nhị phân.
- Mở file để đọc và ghi cho dạng nhị phân.

**Câu 91: Ý nghĩa của hàm `__init__()` trong Python là gì?**

- Khởi tạo một lớp để sử dụng.

- Được gọi khi một đối tượng mới được khởi tạo.
- Khởi tạo và đưa tất cả các thuộc tính dữ liệu về 0 khi được gọi.
- Không có đáp án đúng.

**Câu 92: Cho biết kết quả chương trình sau:**

```
class Foo:
    def printLine(self, line='Python'):
        print(line)
o1 = Foo()
o1.printLine('Java')
```

- Python
- line
- Java
- Java Python

**Câu 93: Cho biết kết quả chương trình sau:**

```
class Point:
    def __init__(self, x = 0, y = 0):
        self.x = x+1
        self.y = y+1
p1 = Point()
print(p1.x, p1.y)
```

- 0 0
- 1 1
- None None
- x y

**Câu 94: Khẳng định nào sau đây là đúng?**

- Trong Python, một toán tử có thể có hoạt động khác nhau tùy thuộc vào toán hạng được sử dụng.
- Bạn có thể thay đổi cách các toán tử hoạt động trong Python.
- `__add()` được gọi khi toán tử '+' được sử dụng.
- Tất cả các đáp trên đều đúng.

**Câu 95: Cho biết kết quả chương trình sau:**

```
class Point:
    def __init__(self, x = 0, y = 0):
        self.x = x
        self.y = y
    def __sub__(self, other):
        x = self.x + other.x
```

```

        y = self.y + other.y
        return Point(x,y)
p1 = Point(3, 4)
p2 = Point(1, 2)
result = p1-p2
print(result.x, result.y)

```

- 2 2
- 4 6
- 0 0
- 1 1

**Câu 96: Khẳng định nào là đúng về chương trình dưới đây?**

```

class A:
    def __init__(self):
        self.a = 1
        self.__b = 1
    def getY(self):
        return self.__b

obj = A()
obj.a = 45
print(obj.a)

```

- Chương trình có lỗi xảy ra vì ' \_\_b ' là thuộc tính private, không thể truy cập được từ bên ngoài lớp.
- Chương trình chạy bình thường và kết quả được in ra là 1.
- Chương trình có lỗi xảy ra vì ' a ' là thuộc tính private, không thể truy cập được từ bên ngoài lớp.
- Chương trình chạy bình thường và kết quả được in ra là 45.

**Câu 97: Kết quả nào là output của đoạn code dưới đây?**

```

try:
    print("throw")
except:
    print("except")
finally:
    print("finally")

```

- finally  
throw
- finally  
except



- except  
finally
- throw  
finally

**Câu 98: Kết quả nào là output của đoạn code dưới đây?**

```
def myfunc():
    try:
        print('Monday')
    finally:
        print('Tuesday')
myfunc()
```

- Tuesday
- Monday Tuesday
- Tuesday Monday
- Monday

**Câu 99: Đoạn code sau có ý nghĩa gì?**

```
try:
    # đoạn code có thể gây ra lỗi
    pass
except (TypeError, ZeroDivisionError):
    print("Python Quiz")
```

- In ra ' Python Quiz ' nếu có ngoại lệ xảy ra (không quan trọng là ngoại lệ gì).
- In ra ' Python Quiz ' nếu không có ngoại lệ xảy ra.
- In ra ' Python Quiz ' nếu một trong hai ngoại lệ TypeError và ZeroDivisionError xảy ra.
- Chỉ in ra ' Python Quiz ' khi cả hai ngoại lệ TypeError và ZeroDivisionError cùng xảy ra

**Câu 100: Cho biết kết quả chương trình sau?**

```
number = 5.0
try:
    r = 10/number
    print(r)
except:
    print("Oops! Error occurred.")
```

- Oops! Error occurred.

- 2.0
- 2.0 Oops! Error occurred.
- 5.0

**Câu 101: Cho đoạn code sau:**

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

**Cách đúng để khởi tạo lớp Dog ở trên là:**

- Dog.create("Rufus", 3)
- Dog.\_\_init\_\_("Rufus", 3)
- Dog()
- Dog("Rufus", 3)

**Câu 102: Cho biết kết quả chương trình sau:**

```
class Dog:
    def walk(self):
        return "*walking*"
    def speak(self):
        return "Woof!"

class JackRussellTerrier(Dog):
    def speak(self):
        return "Arff!"

bobo = JackRussellTerrier()
print(bobo.walk())
```

- AttributeError: 'JackRussellTerrier' object has no attribute 'walk'
- Woof!
- Arff!
- \*walking\*

**Câu 103: Cho biết kết quả chương trình sau:**

```
class Dog:
    def walk(self):
        return "*walking*"
    def speak(self):
        return "Woof!"
```

```
class JackRussellTerrier(Dog):
    def speak(self):
        return "Arff!"

bobo = JackRussellTerrier()
print(bobo.speak())
```

- \*walking\*
- Arff!
- Có xuất hiện lỗi
- Woof!

**Câu 104: Cho biết kết quả chương trình sau:**

```
class Dog:
    def walk(self):
        return "*walking*"
    def speak(self):
        return "Woof!"

class JackRussellTerrier(Dog):
    def talk(self):
        return super().speak()

bobo = JackRussellTerrier()
print(bobo.talk())
```

- \*walking\*
- super
- Có xuất hiện lỗi
- Woof!

**Câu 105: Cho biết kết quả chương trình sau:**

```
class A:
    def __init__(self, x = 2, y = 3):
        self.x = x
        self.y = y

    def __str__(self):
        return "A"

    def __eq__(self, num):
        return self.x * self.y == num.x * num.y

def main():
    a = A(1, 2)
```

```
b = A(2, 1)
print(a == b)

main()
```

- True
- False
- 2
- 1

**Câu 106: Cho biết kết quả chương trình sau:**

```
class A:
    def __str__(self):
        return "A"

class B(A):
    def __init__(self):
        super().__init__()

class C(B):
    def __init__(self):
        super().__init__()

def main():
    b = B()
    a = A()
    c = C()
    print(a, b, c)

main()
```

- B B B
- A B C
- C B A
- A A A