# The Last Lecture

In Class Quiz Game

- form teams now! (max 7 members, min 5 members)

```
def foo(k):
    k = [1]
q = [0]
foo(q)
print(q)
```

a) [0].

b) [1].

c) [1, 0].

d) [0, 1].

[1] as q is mutable list
And copy is not made when
Calling as argument,
So it gets modified

What is the output of the following code?

```
def foo(x):
    x[0] = ['def']
    x[1] = ['abc']
    return id(x)
q = ['abc', 'def']
print(id(q) == foo(q))
```

a) True
b) False
c) None
d) Error

True

```
x = [i**+1 for i in range(3)]; print(x);
```

a) [0, 1, 2].

b) [1, 2, 5].

c) error, **+ is not a valid operator

d) error, ';' is not allowed

^ View Answer

Answer: a

```
print([i+j for i in "abc" for j in "def"])
```

a) ['da', 'ea', 'fa', 'db', 'eb', 'fb', 'dc', 'ec', 'fc'].
b) [['ad', 'bd', 'cd'], ['ae', 'be', 'ce'], ['af', 'bf', 'cf']].
c) [['da', 'db', 'dc'], ['ea', 'eb', 'ec'], ['fa', 'fb', 'fc']].
d) ['ad', 'ae', 'af', 'bd', 'be', 'bf', 'cd', 'ce', 'cf'].

^ View Answer

Answer: d

```
print([if i%2==0: i; else: i+1; for i in range(4)])
```

a) [0, 2, 2, 4].
b) [1, 1, 3, 3].
c) error
d) none of the mentioned

^ View Answer

Answer: c

Read the information given below carefully and write a list comprehension such that the output is: ['e', 'o']

```
w="hello"
v=('a', 'e', 'i', 'o', 'u')
```

a) [x for w in v if x in v]
b) [x for x in w if x in v]
c) [x for x in v if w in v]
d) [x for v in w for x in w]

^ View Answer

Answer: b

Which of the following statements is wrong about inheritance?

a) Protected members of a class can be inherited

b) The inheriting class is called a subclass

c) Private members of a class can be inherited and accessed

d) Inheritance is one of the features of OOP

c

# Output of following code

(a) 1
(b) 2
(c) 3

Output is 2, due
To method
resolution order

```
>>> class A:
...     id=1
...
>>> class B:
...     id=2
...
>>> class C:
...     id=3
...
>>> class M(B,C,A):
...     pass
...
>>> m=M()
>>> m.id
```

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

a) 0 1
b) 0 0
c) Error because class B inherits A but variable x isn't inherited
d) Error because when object is created, argument must be passed like Derived_Test(1)

c

```
class A():
    def disp(self):
        print("A disp()")
class B(A):
    pass
obj = B()
obj.disp()
```

d

a) Invalid syntax for inheritance

b) Error because when object is created, argument must be passed

c) Nothing is printed

d) A disp()

```
class A:
    def __init__(self, x= 1):
        self.x = x
class der(A):
    def __init__(self,y = 2):
        super().__init__()
        self.y = y
def main():
    obj = der()
    print(obj.x, obj.y)
main()
```

a) Error, the syntax of the invoking method is wrong
b) The program runs fine but nothing is printed
c) 1 0
d) 1 2

1 2
Default values of x and y
Are printed

```
class A:
    def __str__(self):
        return '1'
class B(A):
    def __init__(self):
        super().__init__()
class C(B):
    def __init__(self):
        super().__init__()
def main():
    obj1 = B()
    obj2 = A()
    obj3 = C()
    print(obj1, obj2,obj3)
main()
```

a) 1 1 1
b) 1 2 3
c) '1' '1' '1'
d) An exception is thrown
^ View Answer

Answer: a
Explanation: The super().__init__() in the subclasses has been properly invoked and none of other subclasses return any other value. Hence 1 is returned each time the object is created and printed.

```
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)
```

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

^ View Answer

Answer: c
Explanation: Since 5*2==2*5, True is printed. Execute it in the Python shell to verify.

```
def fact(num):
    if num == 0:
        return 1
    else:
        return ____________________
```

a

a) num*fact(num-1)

b) (num-1)*(num-2)

c) num*(num-1)

d) fact(num)*fact(num-1)

```
def f1():
    x=15
    print(x)
x=12
f1()
```

a) Error

b) 12

c) 15

d) 1512

^ View Answer

Answer: c

```
def f1():
    global x
    x+=1
    print(x)
x=12
print("x")
```

a) Error
b) 13
c) 13
x
d) x

View Answer

Answer: d

```
def f1(x):
    global x
    x+=1
    print(x)
f1(15)
print("hello")
```

a) error
b) hello
c) 16
d) 16
hello

^ View Answer

Answer: a

9. What is the output of the code shown below?

```
def f(p, q, r):
    global s
    p = 10
    q = 20
    r = 30
    s = 40
    print(p,q,r,s)
p,q,r,s = 1,2,3,4
f(5,10,15)
```

a) 1 2 3 4
b) 5 10 15 4
c) 10 20 30 40
d) 5 10 15 40

^ View Answer

Answer: c

4. Is the following code valid?

```
try:
    # Do something
except:
    # Do something
else:
    # Do something
```

a) no, there is no such thing as else
b) no, else cannot be used with except
c) no, else must come before except
d) yes

^ View Answer

Answer: d

```
int a = 0;
for (i = 0; i < N; i++) {
    for (j = N; j > i; j--) {
        a = a + i + j;
    }
}
```

**Options:**

1. O(N)
2. O(N*log(N))
3. O(N * Sqrt(N))
4. O(N*N)

Output:

```
4. O(N*N)
```

```
int i, j, k = 0;
for (i = n / 2; i <= n; i++) {
    for (j = 2; j <= n; j = j * 2) {
        k = k + n / 2;
    }
}
```

**Options:**

1. O(n)
2. O(nLogn)
3. O(n^2)
4. O(n^2Logn)

Output:

```
2. O(nLogn)
```

## Array Sorting Algorithms: Mergesort

Time (Best)

- O(nk)
- $O(n^2)$
- O(n log(n))
- O(1)
- O(n + k)
- O(log(n))

nlog(n)

## Data Structure Operations: Array (unsorted)

Search (Worst)

- O(1)
- O(n)
- O(log(n))
- O(n log(n))
- O(m)
- O(mn)

O(n)

## Array Sorting Algorithms: Bubble Sort

Time (Worst)

- ○ O(nk)
- ○ $O(n^2)$
- ○ O(n log(n))
- ○ O(1)
- ○ O(n + k)
- ○ O(log(n))

O(n^2)

## Data Structure Operations: Singly-Linked List

Insertion (Average)

- O(1)
- O(n)
- O(log(n))
- O(n log(n))
- O(m)
- O(mn)

O(n)

## Data Structure Operations: Binary Search Tree

Insertion (Worst)

- O(1)
- O(n)
- O(log(n))
- O(n log(n))
- O(m)
- O(mn)

O(n)

7. What is the time complexity of pre-order traversal in the iterative fashion?

a) O(1)

b) O(n)

c) O(logn)

d) O(nlogn)

O(n)

What is the output when we execute list(“hello”)?

a) [‘h’, ‘e’, ‘l’, ‘l’, ‘o’].

b) [‘hello’].

c) [‘llo’].

d) [‘olleh’].

a)

To shuffle the list(say list1) what function do we use ?

a) list1.shuffle()

b) shuffle(list1)

c) random.shuffle(list1)

d) random.shuffleList(list1)

c)

# Output of following code

```
def foo(a:int)->None:
    print(type(a))

>>> foo('1')
```

(a) <class 'int'>
(b) <class 'str'>
(c) Error

b) as a:int just hint to the user, not restricts a to be an integer

What is the type of sys.argv?

a) set

b) list

c) tuple

d) string

b)

What is the value stored in sys.argv[0]?

a) null

b) you cannot access it

c) the program's name

d) the first argument

c)

How are default arguments specified in the function heading?

a) identifier followed by an equal to sign and the default value

b) identifier followed by the default value within back-ticks (“)

c) identifier followed by the default value within square brackets ([])

d) identifier

a)

```
def fun(n):
    if (n > 100):
        return n - 5
    return fun(fun(n+11));

print(fun(45))
```

a) 50
b) 100
c) 74
d) Infinite loop

30 Second

View Answer

Answer: b
Explanation: The fun(fun(n+11)) part of the code keeps executing until the value of n becomes greater than 100, after which n-5 is returned and printed.

```
def a(n):
    if n == 0:
        return 0
    elif n == 1:
        return 1
    else:
        return a(n-1)+a(n-2)
for i in range(0,4):
    print(a(i),end=" ")
```

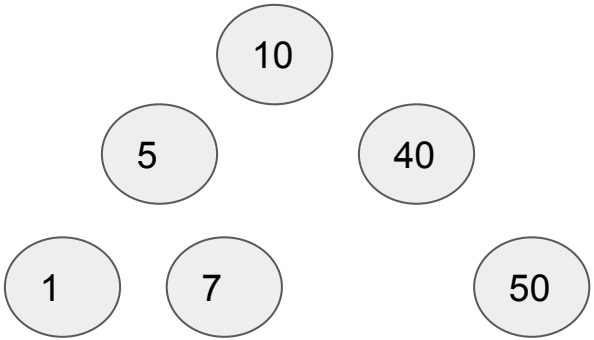
a) 0 1 2 3
b) An exception is thrown
c) 0 1 1 2 3
d) 0 1 1 2
^ View Answer

Answer: d
Explanation: The above piece of code prints the Fibonacci series.

30 Second

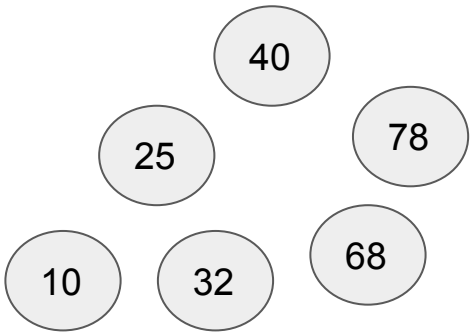
Given the following pre-order traversal, construct the binary search tree

10, 5, 1, 7, 40, 50



Given the following post-order traversal, construct the binary search tree

10, 32, 25, 68, 78, 40



Given the following in-order traversal, construct the binary tree

5,8,9,10,28,30