

What is the output of the expression `re.split(r'[aeiou]', 'abcdefghij')`?  
`['', 'bcd', 'fgh', 'j']`

What does seek method of a file object do?  
Moves the current file position to a different location at a defined offset

Which of the following methods of a match object, `mo`, is used to view the grouped portions of match in the form of a tuple  
`mo.groups()`

Which of the following syntax is used to name a grouped portion of a match?  
`?P<group_name>`

Which of the following methods of a match object, `mo`, is used to view the named group portions of match in the form of a dictionary  
`mo.groupdict()`

What does the search method of re module do?  
it matches the pattern at any position of the string

Which of the following modules support regular expressions in python?  
`re`

In a match found by a defined pattern, how to group various portions of a match  
Using paranthesis, `()`

What does the match method of re module do?  
it matches the pattern at the start of the string

Which of the following expression is used to compile the pattern `p`?  
`re.compile(p)`

=====

Django web frame work uses SQLAlchemy as Object relational mapper. State True or False  
False

pyodbc is an open source Python module that makes accessing ODBC databases simple. State True or False  
True

Which of the following method is used to fetch all the rows of a query result ?  
`fetchall`

Which of the following method is used to fetch next one row of a query result ?  
`fetchone`

Which of the following package provides the utilities to work with postgresQL database?  
`psycopg2`

Which of the following package provides the utilities to work with MySQLDB database?  
`MySQLdb`

While using Object relation mappers for database connectivity, a table is treated as ?  
Class

Which of the following package provides the utilities to work with mongodb database?  
`pymongo`

Which of the following package provides the utilities to work with Oracle

database?  
cx\_Oracle

Which of the following method is used to insert multiple rows at a time in  
sqlite3 database?  
executemany

=====

What is the output of the following code ?

```
def outer(x, y):  
    def inner1():  
        return x+y  
  
    def inner2(z):  
        return inner1() + z  
  
    return inner2
```

```
f = outer(10, 25)
```

```
print(f(15))
```

50

What is the output of the following code ?

```
def outer(x, y):  
    def inner1():  
        return x+y  
  
    def inner2():  
        return x*y  
  
    return (inner1, inner2)
```

```
(f1, f2) = outer(10, 25)
```

```
print(f1())
```

```
print(f2())
```

35

250

A Closure is always a function. State True or False  
True

What is the output of the following code ?

```
v = 'Hello'
```

```
def f():  
    v = 'World'  
    return v
```

```
print(f())
```

```
print(v)
```

World

Hello

What is the output of the following code ?

```
def multipliers():
    return [lambda x : i * x for i in range(4)]

print([m(2) for m in multipliers()])

[6,6,6,6]
```

A Closure does not hold any data with it. State True or False  
False

What is the output of the following code ?

```
def f(x):
    return 3*x

def g(x):
    return 4*x

print(f(g(2)))
```

24

Which of the following are true about functions in python ?

A(x:12, y:3)

=====

What is the output of the following code ?

```
def decorator_func(func):
    def wrapper(*args, **kwargs):
        return func(*args, **kwargs)
    wrapper.__name__ = func.__name__
    return wrapper
```

```
@decorator_func
def square(x):
    return x**2
```

```
print(square.__name__)
```

wrapper

Which of the following is true about decorators ?  
Decorator can be chained

What is the output of the following code ?

```
def bind(func):
    func.data = 9
    return func
```

```
@bind
def add(x, y):
    return x + y
```

```
print(add(3, 10))
print(add.data)
```

13  
9

What is the output of the following code ?

```
def star(func): def inner(args, **kwargs): print("'" * 3) func(args, **kwargs)
print("'" * 3) return inner

def percent(func): def inner(*args, **kwargs): print("%" * 3) func(*args,
**kwargs) print("%" * 3) return inner

@star @percent def printer(msg): print(msg) printer("Hello")

***
%%%
Hello
%%%
***
```

What is the output of the following code ?

```
from functools import wraps

def decorator_func(func):
    @wraps(func)
    def wrapper(*args, **kwargs):
        return func(*args, **kwargs)
    return wrapper

@decorator_func
def square(x):
    return x**2

print(square.__name__)

square
```

What is the output of the following code ?

```
def smart_divide(func): def wrapper(*args): a, b = args if b == 0: print('oops!
cannot divide') return

    return func(*args)
return wrapper
@smart_divide def divide(a, b): return a / b

print(divide.name) print(divide(4, 16))

print(divide(8,0))

wrapper
0.25
oops! cannot divide
None
```

Classes can also be decorated, if required, in Python. State True or False  
True

What is the output of the following code ?

```
class A:

    def __init__(self, val):
        self.x = val
```

```

@property
def x(self):
    return self.__x

    @x.setter
    def x(self, val):
        self.__x = val

    @x.deleter
    def x(self):
        del self.__x

a = A(7)
del a.x
print(a.x)

```

Attribute Error

Which of the following method definitions can a Descriptor have ?  
any of `__get__` , `__set__` , `__delete__`

What is the output of the following code ?

```

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

    @property
    def z(self):
        return self.x + self.y

a = A(10, 15)
b = A('Hello', '!!!')
print(a.z)
print(b.z)

```

25  
Hello!!!

If a property named temp is defined in a class, which of the following decorator statement is required for deleting the temp attribute ?  
`@temp.deleter`

If a property named temp is defined in a class, which of the following decorator statement is required for setting the temp attribute ?  
`@temp.setter`

What is the output of the following code ?

```

class A:

    def __init__(self, value):
        self.x = value

    @property
    def x(self):
        return self.__x

    @x.setter
    def x(self, value):
        if not isinstance(value, (int, float)):
            raise ValueError('Only Int or float is allowed')

```

```
        self.__x = value
```

```
a = A(7)
a.x = 'George'
print(a.x)
```

ValueError

What is the output of the following code ?

```
class A:

    def __init__(self, x):
        self.__x = x

    @property
    def x(self):
        return self.__x
```

```
a = A(7)
a.x = 10
print(a.x)
```

AttributeError

Which of the following is true about property decorator ?

property decorator is used either for getting, setting or deleting an attribute

=====

What is the output of the following code ?

```
class A:

    @staticmethod
    def m1(self):
        print('Static Method')

    @classmethod
    def m1(self):
        print('Class Method')
```

A.m1()

Class Method

Static Method is bound to Objects and also the Class. State True or False  
False

What is the output of the following code ?

```
class A:

    @classmethod
    def getC(self):
        print('In Class A, method getC.')
```

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

```
b = B()
B.getC()
b.getC()
```

In Class A, method getC

In Class A, method getC

Which of the following decorator function is used to create a class method?  
classmethod

What is the output of the following code ?

```
def s1(x, y):  
    return x*y  
  
class A:  
  
    @staticmethod  
    def s1(x, y):  
        return x + y  
  
    def s2(self, x, y):  
        return s1(x, y)  
  
a = A()  
print(a.s2(3, 7))
```

21

What is the output of the following code ?

```
class A:  
  
    @classmethod  
    def m1(self):  
        print('In Class A, Class Method m1.')  
  
    def m1(self):  
        print('In Class A, Method m1.')  
  
a = A()  
  
a.m1()
```

In Class A, Method m1

Which of the following decorator function is used to create a static method?  
staticmethod

What is the output of the following code ?

```
class A:  
  
    @staticmethod  
    @classmethod  
    def m1(self):  
        print('Hello')
```

A.m1(5)

TypeError

=====

What is the output of the following code ?

```
from abc import ABC, abstractmethod  
  
class A(ABC):
```

```

    @classmethod
    @abstractmethod
    def m1(self):
        print('In class A, Method m1.')

```

```

class B(A):

```

```

    @classmethod
    def m1(self):
        print('In class B, Method m1.')

```

```

b = B()
b.m1()
B.m1()
A.m1()

```

```

In class B, Method m1.
In class B, Method m1.
In class A, Method m1.

```

Which of the following module helps in creating abstract classes in Python?  
abc

What is the output of the following code ?

```

from abc import ABC, abstractmethod

```

```

class A(ABC):

```

```

    @abstractmethod
    def m1(self):
        print('In class A, Method m1.')

```

```

class B(A):

```

```

    def m1(self):
        print('In class B, Method m1.')

```

```

class C(B):

```

```

    def m2(self):
        print('In class C, Method m2.')

```

```

c = C()
c.m1()
c.m2()

```

```

In class B, Method m1.
In class C, Method m2.

```

What is the output of following code ?

```

from abc import ABC, abstractmethod

```

```

class A(ABC):

```

```

    @abstractmethod
    def m1():
        print('In class A, Method m1.')

```

```

    def m2():
        print('In class A, Method m2.')

```

```

class B(A):

```



```
def m2():
    print('In class B, Method m2.')
```

```
b = B()
b.m2()
```

TypeError

Which of the following decorator function is used to create an abstract method  
abstractmethod

What is the output of the following code ?

```
from abc import ABC, abstractmethod
```

```
class A(ABC):
```

```
    @abstractmethod
    @classmethod
    def m1(self):
        print('In class A, Method m1.')
```

```
class B(A):
```

```
    @classmethod
    def m1(self):
        print('In class B, Method m1.')
```

```
b = B()
b.m1()
B.m1()
A.m1()
```

AttributeError

What is the output of following code ?

```
from abc import ABC, abstractmethod
```

```
class A(ABC):
```

```
    @abstractmethod
    def m1():
        print('In class A.')
```

```
a = A()
a.m1()
```

TypeError

What is the output of the following code ?

```
from abc import ABC, abstractmethod
```

```
class A(ABC):
```

```
    @abstractmethod
    def m1(self):
        print('In class A, Method m1.')
```

```
class B(A):
```

```
    @staticmethod
    def m1(self):
```

```
print('In class B, Method m1.')
```

```
b = B()  
B.m1(b)
```

In class B, Method m1.

What is the output of the following code ?

```
from contextlib import contextmanager
```

```
@contextmanager  
def context():  
    print('Entering Context')  
    yield  
    print("Exiting Context")
```

```
with context():  
    print('In Context')
```

Entering Context

In Context

Exiting Context

Which of the following module helps in creating a context manager using decorator contextmanager ?  
contextlib

What does the contex manger do when you are opening a file using with.  
It closes the opened file automatically

ZipFile utility of zipfile module is a context manager. State True or False  
True

Popen of subprocess module is a context manager. State True or False  
True

Which of the following keywords is used to enable a context manager in Python ?  
with

What is the output of the following code?

```
from contextlib import contextmanager
```

```
@contextmanager  
def tag(name):  
    print("<%s>" % name)  
    yield  
    print("</%s>" % name)
```

```
with tag('h1') :  
    print('Hello')
```

<h1>  
Hello  
</h1>

What is the output of the following code ?

```
def stringDisplay():  
    while True:  
        s = yield
```

```
print(s*3)
```

```
c = stringDisplay()  
c.send('Hi!!!')
```

TypeError

What is the output of the following code ?

```
def stringDisplay():  
    while True:  
        s = yield  
        print(s*3)
```

```
c = stringDisplay()  
next(c)  
c.send('Hi!!!')
```

Hi!!!Hi!!!Hi!!!

What is the output of the following code ?

```
def nameFeeder():  
    while True:  
        fname = yield  
        print('First Name:', fname)  
        lname = yield  
        print('Last Name:', lname)
```

```
n = nameFeeder()  
next(n)  
n.send('George')  
n.send('Williams')  
n.send('John')
```

First Name: George  
Last Name: Williams  
First Name: John

Select the most correct statement that differentiates a Generator from a Coroutine

Only Coroutines takes input values

Which of the following methods is used to pass input value to a coroutine  
send

What is the output of the following code ?

```
def stringParser():  
    while True:  
        name = yield  
        (fname, lname) = name.split()  
        f.send(fname)  
        f.send(lname)
```

```
def stringLength():  
    while True:  
        string = yield  
        print("Length of '{}' : {}".format(string, len(string)))
```

```
f = stringLength(); next(f)
```

```
s = stringParser()  
next(s)  
s.send('Jack Black')
```

Length of 'Jack' : 4  
Length of 'Black' : 5

A Coroutine is a generator object. State True or False

True

=====

What is the output of the expression `re.sub(r'[aeiou]', 'X', 'abcdefghij')` ?  
XbcdXfghXj

Which of the following command is used to read the next line from a file using the file object `fo`?

`fo.readline()`

Which of the following command is used to read `n` number of bytes from a file using the file object `fo`?

`fo.read(n)`