

## Question 1

```
In [ ]: #The correct syntax for a do-while loop in C# is:  
Answer: do {statement;} while (condition);
```

```
In [ ]:
```

## Question 2

```
In [ ]: # Which of the following is not a correct way to declare a list in python  
#Answer: list = [1,"two",3.1415,[l,i,s,t]]
```

```
In [ ]:
```

## Question 3

```
In [ ]: # A null value is  
#answer: the absence of a Value
```

```
In [ ]:
```

## Question 4

```
In [ ]: #All python classes should have a function called __init__()  
#answer: False
```

```
In [ ]:
```

## Question 5

```
In [ ]: #An advantage of object orientated programming is
```

It hides complexity

We don't need to know how a library works

It streamlines the programming process

#Answer: All of the above

```
In [ ]:
```

## Question 6

```
In [ ]: #Python uses __0__ based indexing  
#Answer: 0
```

```
In [ ]:
```

## Question 7

```
In [ ]: #Which of the following statements is correct about encapsulation in C#?  
  
#Answer: Encapsulation, in object oriented programming methodology, prevents access
```

```
In [ ]:
```

## Question 8

```
In [ ]: #Python variables are declared using which python keyword  
#Answer: var
```

```
In [ ]:
```

## Question 9

```
In [ ]: #Which of the following are the standard python data types  
#Answer: Numbers,Strings, List,Tuple, Dictionary, Boolean
```

```
In [ ]:
```

## Question 10

```
In [ ]: #Which of the following is NOT an Arithmetic operator in C#.NET?  
#Answer: **
```

```
In [ ]:
```

## Question 11

```
In [ ]: #Which of the following is not a valid python declaration  
#Answer: finally = "Done"
```

```
In [ ]:
```

## Question 12

```
In [ ]: #Which datatype should be more preferred for storing a simple number like 35 to imp  
#Answer: int
```

```
In [ ]:
```

## Question 13

```
In [ ]: #Reference is a ____  
#Answer: Copy of class creating by an existing instance.
```

```
In [ ]:
```

## Question 14

```
In [1]: a = 0  
while a < 5:  
    a = a + 1  
    print (a)
```

```
1  
2  
3  
4  
5
```

```
In [ ]:
```

## Question 15

```
In [ ]: #Although there are not many wrong ways to name a variable, by convention the best  
#Answer: descriptive name that reflects its purpose. For example "student_id" or "s
```

```
In [ ]:
```

## Question 16

```
In [ ]: #Assuming the use of mySQL and the sqlite3 python library what is the correct state  
#Answer: conn = sqlite3.connect('databaseName.sqlite')
```

```
In [ ]:
```

## Question:17

```
In [ ]: #The http portion of the link http://www.w3.org/Consortium/mission.html is called t  
#Answer: Scheme
```

```
In [ ]:
```

## Question 18

```
In [ ]: #CLR is the .Net equivalent of ____  
#Answer: Java Virtual machine
```

```
In [ ]:
```

## Question 19

```
In [ ]: #An HTML file is referred to as the  
#Answer: Source file
```

```
In [ ]:
```

## Question 20

```
In [ ]: #Which of the following keyword is used for including the namespaces in the program  
#Answer: using
```

```
In [ ]:
```

## Question 21

```
In [ ]: #__init__(): Constructor that takes as input a pair of Point objects that represent  
  
Length(): returns the length if the segment  
  
Slope() returns the slope of the segment of none if the slope is unbounded  
  
>>> p1 = Point(3,4)  
  
>>> p2 = Point()  
  
>>> s = Segment(p1,p2)  
  
>>> s.length()  
  
5.0  
  
>>> s.slope()  
  
0.75
```

```
In [3]: import math  
  
class Point:  
    def __init__(self, x=0, y=0):  
        self.x = x  
        self.y = y
```

```

class Segment:
    def __init__(self, p1, p2):
        self.p1 = p1
        self.p2 = p2

    def length(self):
        return math.sqrt((self.p2.x - self.p1.x)**2 + (self.p2.y - self.p1.y)**2)

    def slope(self):
        if self.p2.x - self.p1.x == 0:
            return None # slope is undefined (vertical line)
        return (self.p2.y - self.p1.y) / (self.p2.x - self.p1.x)

# the Test for implementation
p1 = Point(3, 4)
p2 = Point() # Default point at (0, 0)
s = Segment(p1, p2)

print("Length:", s.length()) # Output: 5.0
print("Slope:", s.slope())   # Output: 0.75

```

Length: 5.0  
Slope: 1.3333333333333333

In [ ]:

## Question 22

In [ ]:

```

In [14]: # Import Libraries
import sqlite3

# Connect to the database
conn = sqlite3.connect(':memory:')
cur = conn.cursor()

# Create Weather table and insert data
cur.execute('''CREATE TABLE Weather (
                id INTEGER PRIMARY KEY,
                recordDate DATE,
                temperature INTEGER
            )''')

weather_data = [
    (1, '2015-01-01', 10),
    (2, '2015-01-02', 25),
    (3, '2015-01-03', 20),
    (4, '2015-01-04', 30)
]

cur.executemany('INSERT INTO Weather VALUES (?, ?, ?)', weather_data)

# SQL query to find IDs of dates with higher temperatures compared to previous date
sql_query = '''
    SELECT w1.id
    FROM Weather w1
    JOIN Weather w2 ON w1.recordDate = date(w2.recordDate, '-1 day')
    WHERE w1.temperature > w2.temperature
'''

```

```
# To execute the SQL query
cur.execute(sql_query)

# To Fetch and display the result
result = cur.fetchall()
print("IDs of dates with higher temperatures compared to previous dates:")
for row in result:
    print(row[0])
```

IDs of dates with higher temperatures compared to previous dates:  
2

In [ ]:

## Question 23

```
In [32]: def f(indentation, lines):
          for i in range(lines):
              if i == 0 or i == lines - 1:
                  print(" " * indentation + "*" * (i + 1))
              else:
                  print(" " * indentation + "*")
                  for j in range(i):
                      print(" " * (indentation + j + 2) + "*")

          # Test the function
          f(0, 1)
          print('\n')
          f(0, 2)
          print('\n')
          f(0, 4)
```

\*

\*

\*\*

\*

\*

\*

\*

\*

\*

\*\*\*\*

In [ ]: