

Notations :

- 1.Options shown in green color and with ✓ icon are correct.
- 2.Options shown in red color and with ✗ icon are incorrect.

Change Font Color :	No
Change Background Color :	No
Change Theme :	No
Help Button :	No
Show Reports :	No
Show Progress Bar :	No

Programming in Python

Section Id :	640653139205
Section Number :	1
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	17
Number of Questions to be attempted :	17
Section Marks :	50
Display Number Panel :	Yes
Section Negative Marks :	0
Group All Questions :	No
Section Maximum Duration :	0
Section Minimum Duration :	0
Section Time In :	Minutes
Maximum Instruction Time :	0
Sub-Section Number :	1
Sub-Section Id :	640653389883
Question Shuffling Allowed :	No

Question Number : 1 Question Id : 6406532135748 Question Type : MCQ

Correct Marks : 0

Question Label : Multiple Choice Question

THIS IS QUESTION PAPER FOR THE SUBJECT "FOUNDATION LEVEL : PROGRAMMING IN

PYTHON (COMPUTER BASED EXAM)"

ARE YOU SURE YOU HAVE TO WRITE EXAM FOR THIS SUBJECT?

CROSS CHECK YOUR HALL TICKET TO CONFIRM THE SUBJECTS TO BE WRITTEN.

(IF IT IS NOT THE CORRECT SUBJECT, PLS CHECK THE SECTION AT THE [TOP](#) FOR THE SUBJECTS REGISTERED BY YOU)

Options :

6406536956266. ✓ Yes

6406536956267. ✗ No

Sub-Section Number :

2

Sub-Section Id :

640653389884

Question Shuffling Allowed :

Yes

Question Number : 2 Question Id : 6406532135749 Question Type : MCQ

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the following recursive functions:

```
def factorial(n):  
    if n == 0:  
        return 1  
    return n * factorial(n - 1)  
  
def last_digit(n):  
    return n % 10
```

Which of the following function calls returns the last digit of 5! ?

Options :

6406536956268. ✓ `last_digit(factorial(5))`

6406536956269. ✗ `factorial(last_digit(5))`

6406536956270. ✗ `last_digit(5) * factorial(5)`

6406536956271. ✗ `factorial(5 % 10)`

Question Number : 3 Question Id : 6406532135750 Question Type : MCQ

Correct Marks : 3

Question Label : Multiple Choice Question

Consider the following class:

```
class BankAccount:
    total_accounts = 0

    def __init__(self, name, balance):
        self.name = name
        self.balance = balance
        BankAccount.total_accounts += 1

    def deposit(self, amount):
        self.balance += amount
        return self.balance

    def withdraw(self, amount):
        if amount <= self.balance:
            self.balance -= amount
            return True
        return False

acc1 = BankAccount('Priya', 1000)
acc2 = BankAccount('Rahul', 500)
acc1.deposit(500)
x = acc2.withdraw(200)
y = acc2.balance
```

What are the values of `BankAccount.total_accounts`, `x`, and `y` respectively?

Options :

6406536956272. ✖ 1, False, 300

6406536956273. ✖ 2, False, 500

6406536956274. ✔ 2, True, 300

6406536956275. ✖ 3, True, 300

Question Number : 4 Question Id : 6406532135751 Question Type : MCQ

Correct Marks : 3

Question Label : Multiple Choice Question

What will be the output of the following Python code?

```
colors1 = 'red'
colors2 = 'yellow'
result = []
for i in range(len(colors1)):
    for j in range(len(colors2)):
        if colors1[i] != colors2[j]:
            result.append(colors1[i])
            break
        else:
            continue
print(result)
```

Options :

`['r', 'e', 'd']`

6406536956276. ✓

`['r', 'd']`

6406536956277. ✗

`['e']`

6406536956278. ✗

`['y', 'e', 'l', 'l', 'o', 'w']`

6406536956279. ✗

Question Number : 5 Question Id : 6406532135752 Question Type : MCQ

Correct Marks : 3

Question Label : Multiple Choice Question

What is the output of the following snippet of code?

```
sentence = 'hello world'
D = {}
for ch in sentence:
    if ch != ' ':
        D[ch] = D.get(ch, 0) + 1
best_ch = ''
best_count = -1
for k in sorted(D.keys()):
    if D[k] > best_count:
        best_count = D[k]
        best_ch = k
print(best_ch)
```

Options :

h

6406536956280. ✖

l

6406536956281. ✔

o

6406536956282. ✖

w

6406536956283. ✖

Question Number : 6 Question Id : 6406532135753 Question Type : MCQ

Correct Marks : 3

Question Label : Multiple Choice Question

What is the output of the following code?

```
total = 0
for i in range(1, 6):
    if i % 2 == 0:
        total += i
    elif i % 3 == 0:
        total -= i
    else:
        total += i * 2
print(total)
```

Options :

- 6406536956284. ✓ 15
- 6406536956285. ✗ 10
- 6406536956286. ✗ 20
- 6406536956287. ✗ 5

Question Number : 7 Question Id : 6406532135754 Question Type : MCQ

Correct Marks : 3

Question Label : Multiple Choice Question

What is the output of the following snippet of code?

```
s = 'machinelearningwithpython'
D = {}
for ch in s:
    D[ch] = D.get(ch, 0) + 1
mch = ''
mcv = -1
for k in sorted(D.keys()):
    if D[k] > mcv:
        mcv = D[k]
        mch = k
print(mch)
```

Options :

i

6406536956288. ✗

n

6406536956289. ✓

6406536956290. ✖

e

6406536956291. ✖

a

Sub-Section Number :

3

Sub-Section Id :

640653389885

Question Shuffling Allowed :

Yes

Question Number : 8 Question Id : 6406532135755 Question Type : MSQ

Correct Marks : 3 Max. Selectable Options : 0

Question Label : Multiple Select Question

`products` is a list of tuples. Each tuple is of the form `(name, price)`.

Select all snippets of code that create a list `expensive` containing the names of products with price greater than 500. You can assume that `products` is already defined.

Sample Input

```
products = [('Keyboard', 1200), ('Mouse', 300), ('Monitor', 5000), ('USB', 200)]
```

Sample Output

```
['Keyboard', 'Monitor']
```

Options :

```
expensive = [name for (name, price) in products if price > 500]
```

6406536956292. ✔

```
expensive = [name for (name, price) in products if price >= 500]
```

6406536956293. ✖

6406536956294. ✔

```
expensive = []
for (name, price) in products:
    if price > 500:
        expensive.append(name)
```

```
expensive = [x for x in products if x[1] == 500]
```

6406536956295. ✖

Question Number : 9 Question Id : 6406532135756 Question Type : MSQ

Correct Marks : 3 Max. Selectable Options : 0

Question Label : Multiple Select Question

Consider the following class:

```
class StudentGrade:
    students = {}

    def __init__(self, name):
        self.name = name
        self.grades = {}
        StudentGrade.students[name] = self

    def add_grade(self, subject, marks):
        self.grades[subject] = marks

    def get_average(self):
        if not self.grades:
            return 0
        return sum(self.grades.values()) // len(self.grades)

s1 = StudentGrade('Priya')
s2 = StudentGrade('Rahul')
s1.add_grade('Math', 90)
s1.add_grade('Science', 85)
s2.add_grade('Math', 75)
```

Which of the following statements are **TRUE**?

Options :

6406536956296. ✓ `StudentGrade.students` contains 2 entries
6406536956297. ✓ `s1.get_average()` returns 87
6406536956298. ✗ `s2.grades` is equal to `{'Math': 75, 'Science': 85}`
6406536956299. ✓ `s1.grades == {'Math': 90, 'Science': 85}` is True
6406536956300. ✗ `StudentGrade.students['Rahul']` is a string

Question Number : 10 Question Id : 6406532135757 Question Type : MSQ

Correct Marks : 3 Max. Selectable Options : 0

Question Label : Multiple Select Question

Consider the following snippet of code:

```
filename = "test.txt"

with open(filename, "w") as f:
    f.write("\n".join((f"Line{i}" for i in range(1,11))))

with open(filename, "r") as f:
    f.readline()
    f.readline()
    print(f.read(5))
    f.readline()
    print(f.read(10))
    f.seek(0)
    print(f.read(5))
    f.readline()
    f.readline()
    print(f.read(5))
```

Select **all** that apply.

Note

- If a value is passed to `f.read`, it reads that many number of characters from

the current position of the file pointer.

- The `f.seek` method moves the position of the file pointer to the given number of chars after the start of the file.

Options :

6406536956301. ✓ The first line of the output is `Line3`.

6406536956302. ✗ The second line of the output is `Line5`.

6406536956303. ✓ The second line of the output is `Line4`.

6406536956304. ✓ There are 5 lines in the output.

6406536956305. ✗ The output contains empty lines.

Sub-Section Number : 4
Sub-Section Id : 640653389886
Question Shuffling Allowed : Yes

Question Number : 11 Question Id : 6406532135758 Question Type : SA Keyboard Layout : Inscript

Correct Marks : 3

Question Label : Short Answer Question

Consider the following Python code snippet:

```
def f_count(lst):
    if not lst:
        return 0
    result = 1 if lst[0].isupper() else 0
    if len(lst) == 1:
        return result
    return result + f_count(lst[1:-1]) + (1 if lst[-1].isupper() else 0)

result = f_count(['A', 'b', 'C', 'd', 'E', 'f', 'G'])
print(result)
```

What is the output of the program?

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Equal

Text Areas : PlainText

Possible Answers :

4

Question Number : 12 **Question Id :** 6406532135759 **Question Type :** SA **Keyboard Layout :** Inscript

Correct Marks : 3

Question Label : Short Answer Question

What is printed?

```
total = 10
count = 0

for val in [2, "10", 5, "abc"]:
    try:
        result = int(val) + 5
        total += result
    except ValueError:
        count += 1
        total -= 3

print(total + count)
```

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Equal

Text Areas : PlainText

Possible Answers :

40

Question Number : 13 **Question Id :** 6406532135760 **Question Type :** SA **Keyboard Layout :** Inscript

Correct Marks : 3

Question Label : Short Answer Question

What is the output of the following snippet?

```
def is_palindrome(s):
    if len(s) <= 1:
        return True
    if s[0] != s[-1]:
        return False
    return is_palindrome(s[1:-1])

numbers = [11, 121, 12, 151, 99]
count = 0
for num in numbers:
    if is_palindrome(str(num)):
        count += 1
print(count)
```

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Equal

Text Areas : PlainText

Possible Answers :

4

Question Number : 14 Question Id : 6406532135761 Question Type : SA Keyboard Layout : Inscript

Correct Marks : 3

Question Label : Short Answer Question

What is the output?

```
data = [10, 20, 30]
accumulated = []
total = 0

for i in range(len(data)):
    accumulated += data[:i+1]
    for x in accumulated:
        total += x

print(total)
```

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Equal

Text Areas : PlainText

Possible Answers :

150

Question Number : 15 **Question Id :** 6406532135762 **Question Type :** SA **Keyboard Layout :** Inscript

Correct Marks : 3

Question Label : Short Answer Question

What is the output of the following code?

```
def nested_call(n):  
    def helper(x):  
        return x * x  
  
    if n == 0:  
        return 0  
    return helper(n) + nested_call(n - 1)  
  
print(nested_call(4))
```

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Equal

Text Areas : PlainText

Possible Answers :

30

Sub-Section Number : 5

Sub-Section Id : 640653389887

Question Shuffling Allowed : No

Question Id : 6406532135763 **Question Type :** COMPREHENSION **Sub Question Shuffling**

Allowed : No **Group Comprehension Questions :** No **Question Pattern Type :** NonMatrix

Question Numbers : (16 to 17)

Question Label : Comprehension

```
d = {'P': 97, 'C': 93, 'B': 98}
total = 0
for key in d:
    total += d[key]
print(total)
```

```
# Hint: max(iterable, key=func) returns the element for which func(element) is maximum.
k = max(d, key=d.get)
print(k)
```

Based on the above program, answer the given subquestions.

Sub questions

Question Number : 16 Question Id : 6406532135764 Question Type : MCQ

Correct Marks : 2

Question Label : Multiple Choice Question

What does the first `print` statement output?

Options :

6406536956311. ✖ 97

6406536956312. ✖ 93

6406536956313. ✔ 288

6406536956314. ✖ 98

Question Number : 17 Question Id : 6406532135765 Question Type : MCQ

Correct Marks : 2

Question Label : Multiple Choice Question

What is printed in the second line?

Options :

6406536956315. ✖ P

6406536956316. ✖ C

6406536956317. ✔ B

6406536956318. ✖ None

Question Id : 6406532135766 Question Type : COMPREHENSION Sub Question Shuffling

Allowed : No Group Comprehension Questions : No Question Pattern Type : NonMatrix

Question Numbers : (18 to 19)

Question Label : Comprehension

```

def f(matrix):
    n = len(matrix)
    for i in range(n):
        temp = matrix[i][i]
        matrix[i][i] = matrix[n - 1 - i][n - 1 - i]
        matrix[n - 1 - i][n - 1 - i] = temp
    return matrix

def g(matrix):
    prod = 1
    n = len(matrix)
    for i in range(n):
        prod *= matrix[i][i]
    return prod

M = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
result = f(M)
print(result)
print(g(result))

```

Based on the above program, answer the given subquestions.

Sub questions

Question Number : 18 Question Id : 6406532135767 Question Type : SA Keyboard Layout : Inscript

Correct Marks : 2

Question Label : Short Answer Question

What is the element at position [0][2] in `result`?

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Equal

Text Areas : PlainText

Possible Answers :

3

Question Number : 19 Question Id : 6406532135768 Question Type : SA Keyboard Layout : Inscript

Correct Marks : 2

Question Label : Short Answer Question

What is the value of the second line of output?

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Equal

Text Areas : PlainText

Possible Answers :

45