

[This question paper contains 15 printed pages]

Your Roll No. :

Sl. No. of Q. Paper : **1356** **I**

Unique Paper Code : 2342011104

Name of the Paper : Object Oriented
Programming using
Python (DSC-I)

Name of the Course : **B.Sc.(H) Computer
Science**

Semester : I

Time : 3 Hours **Maximum Marks : 90**

Instructions for Candidates :

- Write your Roll No. on the top immediately on receipt of this question paper.
- Section - A** is compulsory.
- Attempt any **4** questions from **Section-B**.
- Parts of a question must be answered together.

Section - A
(Compulsory)

30

- (a) Which of the following are invalid identifiers ?
Justify your answer : 2

P.T.O.

- (i) **Course - Name**
 - (ii) **7 wonders**
 - (iii) **Except**
 - (iv) **Riturn**
- (b) Draw a flow chart to find the smallest of three numbers. 4
- (c) Write a function **findFactors ()** that takes two numbers, **number 1** and **number 2** as input parameters and returns a tuple comprising common factors of these two numbers. 4
- For example, if **number 1** is **18** and **number 2** is **30**, then the function should return **(1, 2, 3, 6)**.
- (d) Consider the following Python code segments and determine the output produced on their execution : 4

```
(i) greetWith ="good morning friends"
result = ""
for i in greetWith. title ( ). split ( ) :
    result += (i[:-1] + i [-1]. upper ( )) + " "
print (result, strip ( ))
```

- * Add the complex numbers referred by **C1** and **C2** and assign the result to **C3**.
- * Display the complex number **C3** using the **_str_ () method**.

- (i) The class should support the following methods :

* **_init_** for initializing the real and imaginary parts of the complex numbers.

* **_add_ ()** to overload the + operator, allowing addition of two complex numbers.

* **_str_ ()** that returns the complex number in the form **a + bi** or **a - bi** based on the values of **real** and **imag**.

- (ii) Also write Python statements for the following :

* Create an object **C1** of the **ComplexNumber** class with the values **real = 3** and **imag = 4** to represent the complex number **3+4i**.

* Create an object **C2** of the **ComplexNumber** class with the values **real = 4** and **imag = 5** to represent the complex number **4 + 5i**.

- (ii) Assume that the file '**notes.txt**' does not exist.

try :

f = open ('notes.txt', 'w')

except IOError :

print ('Error occurred while opening')

else :

print ('File opened successfully for writing')

try :

f = open ('notes.txt', 'r')

except IOError :

print ('Error occurred while opening')

else :

Print ('File opened successfully for reading')

- (e) Consider the following dictionary **subjectMarks** representing the subject names and marks obtained in corresponding subject :

5

```
SubjectMarks = {'Chemistry': 90, 'Hindi': 85,
                'Mathematics': 95, 'Physics':
                89,
                'Biology': 92, }
```

Write Python statements for the following :

- (i) Retrieve the marks in 'Mathematics'.
- (ii) Add the contents of the following dictionary to **subjectMarks**:

```
{'Sanskrit' : 78, 'Geography' : 88,
 'Economics': 84}
```

- (iii) How will Python respond when the following statement is executed ?

```
subjectMarks.get('Phylosophy', -1)
```

- (iv) Delete the details of the subject 'Hindi' from the dictionary.
- (v) What should be the updated contents of the dictionary **subjectMarks** after the execution of the following statement :

```
subjectMarks ['Computer Sc'] = 100
```

- (ii) Remove the transport '**Tempo Traveller**' from the set **heavyVehicles**.
- (iii) Determine the set of average weight vehicles which are neither heavy weight nor light weight.
- (iv) Determine the number of **vehicles**.

7. (a) Consider the following function : 5

```
def addition (num1, num2 = 5, num3 = 38) :
    return num1 + num2 + num3
```

What will be the output returned on execution of the following function calls :

- (i) **addition (num2 = 15, num 1 = 47)**
- (ii) **addition (29)**
- (iii) **addition ()**

- (b) Define a class **ComplexNumber** that represents complex numbers and supports basic arithmetic operations using operator overloading. The class should contain the following data members : 10

real – The real part of the complex number

imag – The imaginary part of the complex number

- (i) **players.rfind ('i')**
- (ii) **players.swapcase ()**
- (iii) **players.lstrip ()**
- (iv) **players.endswith ('!!')**
- (v) **players.replace ('Great', 'outstanding')**

6. (a) Write a function that reads the file **report.txt** and copies even numbered lines to file **evenfile.txt** and odd numbered lines to file **oddfile.txt**. For example, 1st, 3rd, 5th, ... lines of the file **report.txt** should be copied to file **oddfile.txt** and 2nd, 4th, 6th, line of the file **report.txt** should be copied to file **evenfile.txt**. Handle all exceptions that can be raised. 7

(b) Consider the following **three** sets : 8

vehicles = {'Bicycle', 'Scooter', 'Car', 'Bike', 'Truck', 'Bus', 'Tempo Traveller', 'Rickshaw'}

heavyVehicles = {'Truck', 'Bus', 'Tempo Traveller'}

lightVehicles = {'Rickshaw', 'Scooter', 'Bike'}

Write Python statements to perform the following operations on the given sets :

- (i) Add the transport **'Bicycle'** to set **lightVehicles**.

- (f) Consider the following Python program that defines a class **Rectangle**. There are some error (s) in the code, indentify them, and rewrite the corrected program : 5

```
class Rectangle
    def _init (self, l, w) :
        self.length = l
        self.breadth = w

    def area ( ) :
        return length * breadth

def main ( ) :
    ob = Rectangle (5, 4)
    print (area ( ))

main ( )
```

- (g) What is the difference between the following statements ? 6

(i) **Y += y + 5 and y = y + 5**

(ii) **y == 10 and y = 10**

(iii) **for j in range (1, 6):** **for j in range (1, 6):**

if (J % 2 == 1): **if (j % 2 == 1):**

continue

break

print (j)

print (j)

Section - B

(Attempt Any Four Questions)

2. (a) Write a function named **seriesSum ()** that takes an integer **n** and **x** as input parameters and returns the sum of the first **n** terms of the following series : 5

$$x^1/1! - x^2/2! + x^3/3! - x^4/4! + \dots \dots \dots x^n/n!$$

- (b) What will be the output produced on execution of the following Python statements ? 5

- (i) `print (9 + 5 * 2** 3 ! = 15// 6 - 2)`
- (ii) `print (64>>2)`
- (iii) `print ('apple' > 'banana' and 'orange' < 'grape')`
- (iv) `print ('Hello' * 2)`
- (v) `print (2 ** 3** 2)`

- (c) Write a function named as **oddWord ()** that takes a string as an input parameter and returns a new string with every word of odd length replaced with the length of the corresponding word. 5

For example : 'I study in Delhi University' is returned as '1, 5 in 5 University'.

5. (a) Write a function named as **printPattern ()** that accepts the number of rows **n** as an input parameter and prints the pattern comprising of **n** rows of the following format (say, for **n = 5**) : 5

```
1
12
123
1234
12345
```

- (b) Write a program that takes a list of numbers as input from the user and creates a list of squares of all the positive even numbers using list comprehension method. 5

For example, if the input list is [1, -2, 3, 4, 5, 6, -7, 8],

The program should produce the output [16, 36, 64].

- (c) Consider the following string : 5

players = "kohli and rohit play great game "

Write the output produced on execution of the following function calls :

- (i) Write a Python code segment to make a shallow copy, named, **copyProducts**, of **productList**.
- (ii) What will be the output produced on execution of the following Python code segment ?

```
copyProducts [1] [1] = 550
copyProducts [2] = ['Smartwatch', 250]
print (productList)
print (copyProducts)
```

- (c) Find all the errors (if any) in the following Python code segments : $3 \times 2 = 6$

- (i) **f = open ('record1', 'r')**
f.write (Weather is great today')
f. close ()
- (ii) **name = 'Mohinder Amarnath'**
name [-5] = 'u'
lastChar = name [len (name) -1]
- (iii) **studentMarks = [1001, 'Rohan', [90, 85, 99, 50, 99]]**
studentMakrs [2] = 95
print (max (studentMarks))

3. (a) Identify the line number where an exception may be raised on execution. Also specify the reason for the exception. 4

```
marksLst = eval ('Input (Enter list of marks in three subjects : ') # Line 1
```

```
maxMarks = int (input ('Enter maximum marks per subject : ')) # Line 2
```

```
marksObtained = marksLst [1] + marksLst [2] + marksLst [3] # Line 3
```

```
result = marksObtained/(3*maxMarks) # Line 4
```

```
print (result) # Line 5
```

- (b) Write a function **isComposite ()** that takes an integer as an input parameter and returns **True** if the number is composite (i.e., not a prime number and greater than 1) and **False** otherwise. For example : 4

For is **Composite (4)**, the function should return **True** since 4 is a composite number.

For is **Composite (7)**, the function should return **False** since 7 is a prime number.

For is **Composite (1)**, the function should return False since 1 is neither prime nor composite.

- (c) Write a program that takes a list of integers as input from the user and generates a corresponding cumulative list where each element in the resultant list at index 'i' is the sum of all integers at index $J \leq i$.

7

For example, consider the following list :

[7, 12, 4, 9]

Output : Cumulative List : [7, 19, 23, 32]

4. (a) Consider the following function **calculateSpeed ()** that calculates speed using the formula **speed = distance/time** : 4

```
def calculateSpeed (distance, time) :
```

```
try :
```

```
    speed = distance/time
```

```
except ZeroDivissionError :
```

```
    print ('ZeroDivisionError')
```

```
except TypeError :
```

```
    print ('TypeError')
```

```
except ValueError :
```

```
    print ('valueError')
```

```
except :
```

```
    print ('An unexpected error occurred.')
```

```
else :
```

```
    print ('Speed :', speed, 'm/s')
```

```
finally :
```

```
    print ('Execution completed.')
```

What will be the output produced on the execution of the following statements ?

(i) **calculateSpeed (100, 0)**

(ii) **calculateSpeed (100, '20')**

- (b) Consider the following list representing product details : 2+3=5

```
productList = [['Laptop', 800], ['Smartphone', 500],  
               ['Tablet', 300]]
```