

S.Y. B.Sc. (IT) : Sem. III
Python Programming
Prelim Question Paper



Time : 2½ Hrs.]

[Marks : 75

- Instructions :**
- (1) All questions are compulsory.
 - (2) Make suitable assumptions wherever necessary and state the assumptions made.
 - (3) Answers to the same questions must be written together.
 - (4) Numbers to the right indicate marks.
 - (5) Draw neat labeled diagrams wherever necessary.
 - (6) Use of Non-programmable calculators is allowed.

1. Attempt the following (any THREE)

[15]

- (a) Python programs are executed by an interpreter. There are two ways to use the interpreter. What are they?
- (b) A triple of numbers (a,b,c) is called a Pythagorean triple if $a^2 + b^2 = c^2$. In this question, you will be given three numbers. You have to output 1 if the three numbers form a Pythagorean triple. Otherwise, you have to output 0.
Note that the inputs may not be given in order: you may have to try all possible orderings of the three numbers to determine whether they form a Pythagorean triple.
- (c) Explain type conversion of variable in Python.
- (d) Explain if...else statement with example.
- (e) What is the difference between interactive mode and script mode in Python?
- (f) Explain the use of break statement in a loop with example.

2. Attempt the following (any THREE)

[15]

- (a) (i) Name the following code and explain the same:

```
def recurse():  
    recurse()
```

- (ii) def countdown(n):

```
    if n <= 0:  
        print ('Blastoff!')  
    else:  
        print( n)  
        countdown(n-1)
```

What will happen if we call this function as seen below?

```
>>> countdown(3)
```

- (iii) Fermat's Last Theorem says that there are no positive integers a, b, and c such that $a^n + b^n = c^n$ for any values of n greater than 2.

Write a function named check_fermat that takes four parameters—a, b, c and n—and that checks to see if Fermat's theorem holds. If n is greater than 2 and it turns out to be true that $a^n + b^n = c^n$ the program should print, "Holy smokes, Fermat was wrong!" Otherwise the program should print, "Fermat's Theorem holds."

Write a function that prompts the user to input values for a, b, c and n, converts them to integers, and uses check_fermat to check whether they violate Fermat's theorem

- (b) (i) Write the output for the following if the variable fruit = 'banana' :

```
>>> fruit[:3]  
>>> fruit[3:]  
>>> fruit[3:3]  
>>> fruit[:]
```

(ii) What is the reason for the error seen here?

```
>>> greeting = 'Hello, world!'
```

```
>>> greeting[0] = 'J'
```

TypeError: 'str' object does not support item assignment

(iii) Explain any 2 methods under strings in Python with suitable examples.

(c) Short note on incremental development.

(d) Explain str.find() function with suitable example.

(e) Write a Python code to check whether the given number is a strong number. Strong Numbers are numbers whose sum of factorial of digits is equal to the original number (Example: 145 = 1! + 4! + 5!).

(f) The Ackermann function, A(m,n), is defined:

$$A(m, n) = \begin{cases} n+1 & \text{if } m=0 \\ A(m-1, 1) & \text{if } m>0 \text{ and } n=0 \\ A(m-1, A(m, n-1)) & \text{if } m>0 \text{ and } n>0 \end{cases}$$

Write a recursive function named ackermann that evaluates Ackermann's function.

3. Attempt the following (any THREE)

[15]

(a) (i) Write the output for the following :

```
(1) >>> a = [1, 2, 3]
```

```
>>> b = [4, 5, 6]
```

```
>>> c = a + b
```

```
>>> print c
```

```
(2) >>> [1, 2, 3] * 3
```

```
(3) >>> t = ['a', 'b', 'c', 'd', 'e', 'f']
```

```
>>> t[1:3] = ['x', 'y']
```

```
>>> print t
```

(ii) Explain any 2 methods under lists in Python with examples.

(iii) Write the function tail(l) to get the following output:

```
>>> letters = ['a', 'b', 'c']
```

```
>>> rest = tail(letters)
```

```
>>> print rest
```

```
['b', 'c']
```

(b) List and explain any five exceptions in Python.

(c) Explain the following statements:

```
(i) >>> a, b = 1, 2, 3
```

```
(ii) >>> addr = 'monty@python.org'
```

```
>>> uname, domain = addr.split('@')
```

```
(iii) >>> t = divmod(7, 3)
```

```
>>> print( t)
```

```
(iv) >>> s = 'abc'
```

```
>>> t = [0, 1, 2]
```

```
>>> zip(s, t)
```

```
(v) for index, element in enumerate('abc'):
```

```
print( index, element)
```

(d) What is tuple in Python? How to create and access it?

(e) Explain open () and close () methods for opening and closing file.

(f) Write short note on Tuple Assignment (Packing and Unpacking)

4. Attempt the following (any THREE)

[15]

(a) Explain the role of the Regular Expressions in the following snippets:

```
(i) >>> p = re.compile('\d+')
```

```
>>> p.findall('12 drummers drumming, 11 pipers piping, 10 lords a-leaping')
```

```
(ii) >>> p = re.compile('ca+t')
>>> p.findall('ct cat caaat caaaaaat caaat ctttt cccttt')
(iii) >>> p = re.compile('ca*t')
>>> p.findall('ct cat caaat caaaaaat caaat ctttt cccttt')
(iv) >>> p = re.compile('a/{1,3}b')
>>> p.findall('a/b a//b a///b a/////b ab')
(v) >>> result=re.findall(r'\w*', 'AV is largest Analytics community of India')
>>> print (result)
```

- (b) Explain Class inheritance in Python with an example.
- (c) (i) How is method overriding implemented in Python?
(ii) Which methods of Python are used to determine the type of instance and inheritance?
- (d) Explain match () function with suitable example.
- (e) What is module? What are the advantages of using module?
- (f) Explain various functions of math module.

5. Attempt the following (any THREE)

[15]

- (a) Write a Python code to show the following types of message boxes:



- (b) Write a Python code to do the following:
 - (i) Create a table EMPLOYEE with columns Id (integer, not null, primary key), AdhaarNo (integer, not null), Empname (not null, varchar), Empdept (not null, varchar) and Empage (not null, integer) in MySQL and insert 5 rows of data.
 - (ii) Update the respective rows to Empdept = 'ERP' where Empage is >=40.
- (c) Explain Checkbutton widget with example.
- (d) Write short note on cursor object in Python.
- (e) What is layout management? Explain Grid manager.
- (f) Short note on Radiobutton creation.



Paper Discussion Schedule for all Subjects

Date	Day	Timing	Centre
21 Oct. 2018	Sunday	8.00 a.m. to 10.00 a.m.	Dadar
21 Oct. 2018	Sunday	1.00 p.m. to 3.00 p.m.	Andheri
21 Oct. 2018	Sunday	3.30 p.m. to 5.30 p.m.	Borivali
21 Oct. 2018	Sunday	1.00 p.m. to 3.00 p.m.	Thane
21 Oct. 2018	Sunday	3.30 p.m. to 5.30 p.m.	Ghatkopar
22 Oct. 2018	Monday	6.00 p.m. to 8.00 p.m.	Nerul