

EXPERIMENT-1

Priyansh Salian C31

Priyansh Salian
2083148
C31

Page : _____
Date : _____

Experiment 1

AIM :- Exploring basics of python like data types (strings, list, array, dictionaries, set, tuples) and control statements.

Subquestion 1

Write a Python program to print the following string in specific format

15 Twinkle, twinkle, little star,
 " How I wonder what you are !"
 Up above the world so high,
 Like a diamond in the sky.
20 Twinkle, 'twinkle', little star
 How I wonder what you are

Theory :-

25 Print Function :- The python print function takes in any no of parameters & prints them out on one line of test.

Syntax :- `print(value(s), sep=' ', end='\n', file = file, flush = flush)`

Ex :-

`print(12, 24, -2)`

Output :- 12 24 -2

Camlin

Parameters

1. sep =

By default print() separates the items by spaces. The optional sep= parameter sets a different separator text.

Ex :- print(12, 31, 24, sep=':')

Output :- 12:31:24

2. end =

By default print() puts a single '\n' after all the items. The optional end= parameter sets a custom string to appear after all the items.

print("Hello \n")

Output :- Hello

3. file :

This optional parameter is an object with a write method. Default: sys.stdout

4. flush :

This optional parameter is a boolean type which specifies if the output is flushed (True) or buffered (False). Default: False

Priyansh Salian

2003148

C 31

Page :

Date :

Print vs Return

Return is the main way for a function to return results back to its caller.

Print is form of function output, most often used to produce text output for the user in the terminal.

CODE:-

```
Ex_1_1.py
1  print(
2
3  '''Twinkle, twinkle, little star,
4      How I wonder what you are!
5          Up above the world so high,
6              Like a diamond in the sky.
7  Twinkle, ' twinkle ', little star,
8      How I wonder what you are'''
9 )
10
11
12 |
```

OUTPUT:-

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS C:\Users\Puru\Desktop\PRIYANSH\College\PYTHON> python -u "c:\Users\Puru\Desktop\PRIYANSH\College\PYTHON\Ex_1_1.py"
Twinkle, twinkle, little star,
    How I wonder what you are!
        Up above the world so high,
            Like a diamond in the sky.
Twinkle, ' twinkle ', little star,
    How I wonder what you are
PS C:\Users\Puru\Desktop\PRIYANSH\College\PYTHON> []
```

2. Program to show output formattting take
two values & display them using
single print function using
• str.format()
• % operator

20 Theory :

String formatting is the process of inserting things in the string dynamically & presenting the string. There are four different ways to perform string formatting.

a. "Old Style" string formatting (% operator)

Strings in Python have a unique built-in operation that can be accessed with the % operator.

Input :- 'Hello, %s' % name

Output :- "Hello, Bob"

Camlin

We are using `%'s` formal specifier here to tell Python where to substitute the value of name, represented as string

b. "New style" String formatting (str.format)

This "new style" string formatting gets rid of the `%` operator special syntax & makes the syntax ~~for~~ & for string formatting more regular
name = "Bob"

Input :- 'Hello, %s'.format(name)

Output :- Hello, Bob

c. String Interpolation / f-strings

This way of formatting strings lets you use embedded Python expressions inside string constants
f'Hello, {name}!'

Hello, Bob

d. Template String

It's a simpler & less powerful mechanism

Ex:- Input: from string import Template

t = Template('Hey, \${name}')

t.substitute(name=name)

Output: Hey, Bob!

CODE:-

```
Ex_1_2.py > ...
1 one =input("Enter the first string ")
2 two =input("Enter the second string ")
3 |
4
5
6 print ("Combined string is using operator is %s %s" % (one, two))
7 print ("Combined string is using str.format() method is {0} {1}".format(one, two))
```

OUTPUT:-

```
PS C:\Users\Puru\Desktop\PRIYANSH\College\PYTHON> python -u "c:\Users\Puru\Desktop\PRIYANSH\College\PYTHON\Ex_1_2.py"
Enter the first string Die with memories
Enter the second string not dreams
Combined string is using operator is Die with memories not dreams
Combined string is using str.format() method is Die with memories not dreams
PS C:\Users\Puru\Desktop\PRIYANSH\College\PYTHON> |
```

3. Program to find leap year using nested
if

Theory :-

Decision making is required when we want to execute a code only if a certain condition is satisfied for this we need if...else block

Syntax of if...else

if test expression:

Body of if

else:

Body of else

This if block will executes the test expression & will execute the body of if only if the test condition is true.

If the condition is False, the body of else is executed. Indentation is used to separate the blocks.

Nested-if statement

We can have an if..elif..else statement inside another if..elif..else statement. This is also called nesting in computing.

programming. Any no of these can be nested inside one another.

Syntax:

if (condition):

Executes when condition is true

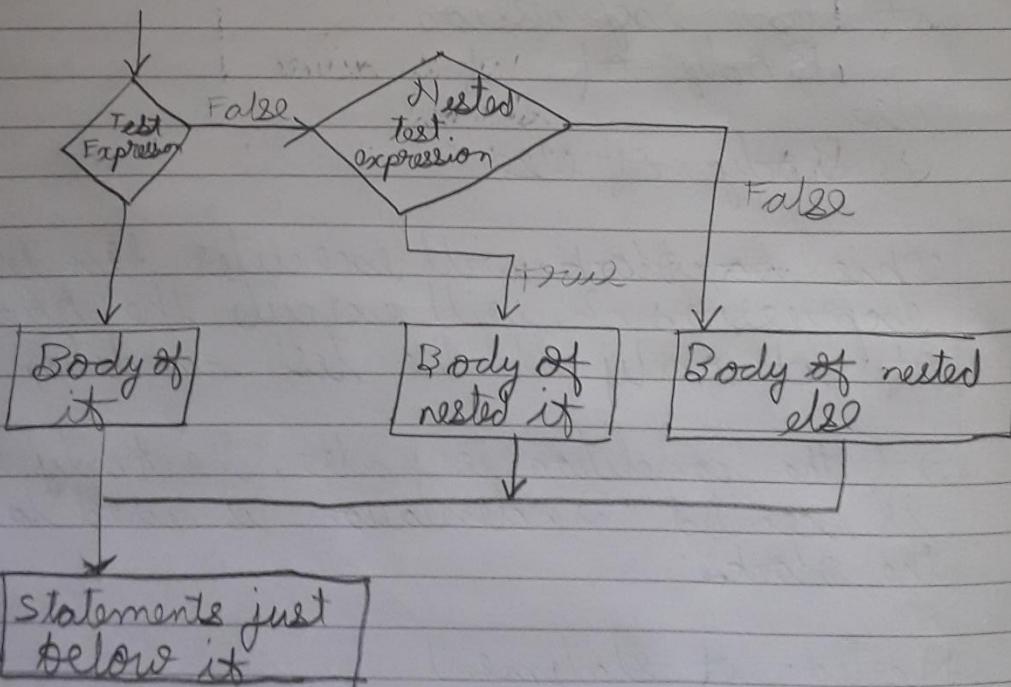
if (condition2)

Executes when condition2 is true

if Block is end here

if Block is end here

Flowchart



For a year to be leap year it has to be divisible by 4, 100 & 400.

CODE:-

```
Ex_1_3.py > ...
1 year=input("Enter the year you want to check if its leap year or not ")
2 year =int(year)
3 if(year%4==0):
4     if(year%400==0 and year%100==0):
5         print("yes it is a leap year")
6     else:
7         print("not a leap year")
8 else:
9     print("not a leap year")
10
```

OUTPUT:-

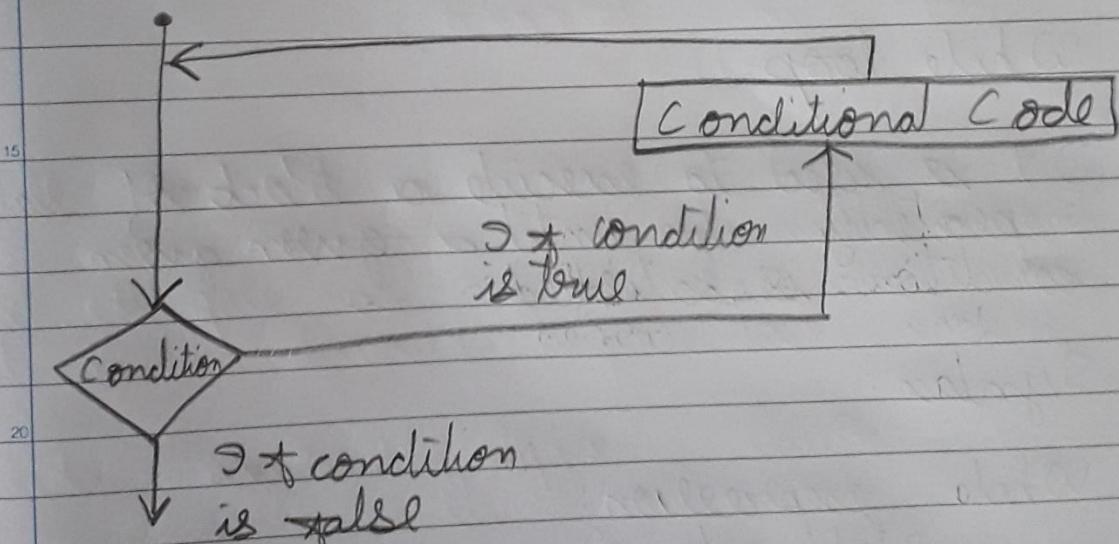
```
PS C:\Users\Puru\Desktop\PRIYANSH\College\PYTHON> python -u "c:\Users\Puru\Desktop\PRIYANSH\College\PYTHON\Ex_1_3.py"
Enter the year you want to check if its leap year or not 1900
not a leap year
PS C:\Users\Puru\Desktop\PRIYANSH\College\PYTHON> python -u "c:\Users\Puru\Desktop\PRIYANSH\College\PYTHON\Ex_1_3.py"
Enter the year you want to check if its leap year or not 2000
yes it is a leap year
PS C:\Users\Puru\Desktop\PRIYANSH\College\PYTHON>
```

4. Program to print all Armstrong
number in range 1 to 1000

5. Theory :-

For printing large no of things we use loops. Loop means statements are executed sequentially.

Flowchart



Types of loops

a. While loop

Repeats a statement or group of statements while a given condition is TRUE. It tests the condition before executing the loop body.

20. for loop

For executes a sequence of statements multiple times and abbreviates the code that manages the loop variable

c. nested loops

You can use one or more loop inside any or another while, for or do-while loop.

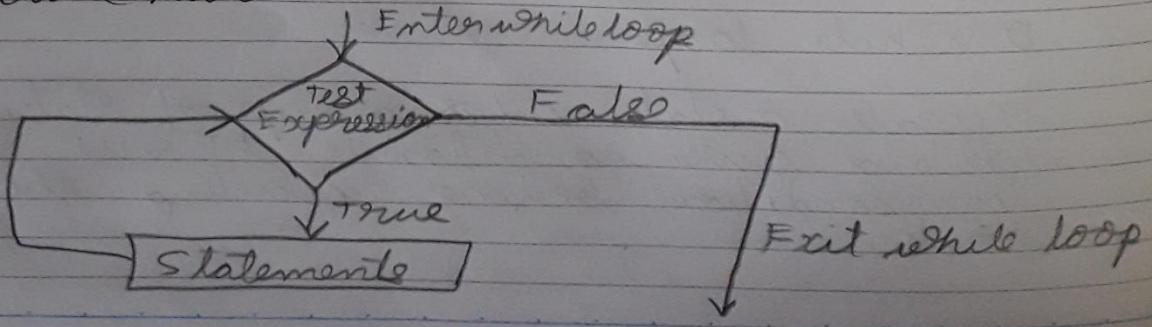
while loop

It is used to execute a block of statements repeatedly until a given condition is satisfied.

Syntax:

while expression:
statements ()

Flowchart



Priyansh Sahai
2003148
C 31

Page :
Date :

Ex :- Amstrong

Amstrong numbers are those numbers whose sum of cube of its digits is equal to number itself.

Ex :- 370 is an amstrong no as

$$3 \times 3 \times 3 + 7 \times 7 \times 7 = 370$$

CODE:-

```
Ex_1_4.py > ...
Search (Ctrl+Shift+F)
2 i=1
3 num=1
4 # for num in range(1,1000):
5 while i<1000:
6     temp=num
7     sum=0
8     while temp>0:
9         digit=temp%10
10        sum=sum+digit*digit*digit
11        temp=temp//10
12
13    if sum==num:
14        print (num)
15    i=i+1
16    num=num+1 |
```

OUTPUT:-

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\Puru\Desktop\PRIYANSH\College\PYTHON> python -u "c:\Users\Puru\Desktop\PRIYANSH\College\PYTHON\Ex_1_4.py"
1
153
370
371
407
PS C:\Users\Puru\Desktop\PRIYANSH\College\PYTHON> |
```

5.10

Program to find Fibonacci series of n terms

Theory :-

15. Fibonacci series is a series where each number is the sum of the two preceding ones. The sequence commonly starts from 0 and 1.

20. Example:-

0, 1, 1, 2, 3, 5, 8, 13, 21, ...
\$

For loop

25. A for loop is used for iterating over a sequence (that is either a list, a tuple, a dictionary, a set, or a string).

30. With the for loop we can execute a set of statements once for each item in a list, tuple, set, etc.

Priyansh Saha
2003148
C 31

Page :
Date :

Ex :- fruits = ["apple", "banana", "cherry"]

for x in fruits:
 print(x)

Output:- apple
 banana
 cherry

Looping through a string

for x in "banana";
 print(x);

Break statement

With break statement we can stop the loop
before it has looped through all the items.

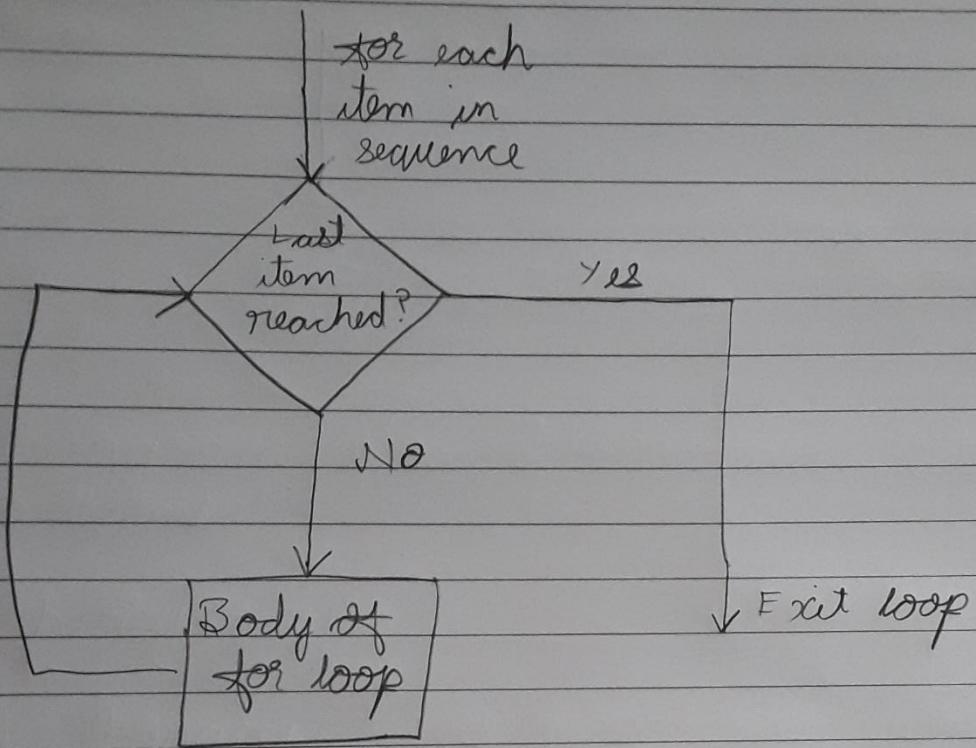
Ex :- fruits = ["apple", "banana", "cherry"]
for x in fruits:
 print(x)
 if x == "banana":
 break

Output:- apple
 banana

Priyansh Salian
2003148
C3)

Page :
Date :

Flow Chart



CODE:-

```
Ex_1_5.py > ...
1 no = int(input("Till what no do you want to print fibonacci series? "))
2
3 n1, n2 = 0, 1
● 4 count = 0
5
6 |
7 if no <= 0:
8     print("Please enter a positive integer")
9
10 elif no == 1:
11     print("Fibonacci sequence upto",no,:)
12     print(n1)
13
14 else:
15     print("Fibonacci sequence:")
16     while count < no:
17         print(n1)
18         nth = n1 + n2
19
20         n1 = n2
21         n2 = nth
22         count += 1
```

OUTPUT:-

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\Puru\Desktop\PRIYANSH\College\PYTHON> python -u "c:\Users\Puru\Desktop\PRIYANSH\College\PYTHON\Ex_1_5.py"
Till what no do you want to print fibonacci series? 10
Fibonacci sequence:
0
1
1
2
3
5
8
13
21
34
PS C:\Users\Puru\Desktop\PRIYANSH\College\PYTHON>
```

6. Program on pattern

20	A	* * * * *	1
	B B	* * * *	1 2 1
	C C C	* * *	1 2 3 2 1
	D D D D	* *	1 2 3 4 3 2 1
	E E E E	*	1 2 3 4 5 4 3 2 1

25

*
* *
* * *
* * * *
* * * * *

30

CODE:-

```
Ex_1_6.py > ...
1  design = [5,4,3,2,1]
2  str=""
3  for i in design:
4      for j in range(i):
5          str = str + "*"
6
7
8
9      print(str)
10     str=""
11
12
13
14 design=[1,2,3,4,5]
15 str=""
16 for i in design:
17
18     for k in range(5-i):
19         str=str+ " "
20
21
```

Ex_1_6.py > ...

```
● 14  design=[1,2,3,4,5]
15  str=""
16  for i in design:
17
18  for k in range(5-i):
19      str=str+" "
20
21
22  for j in range(i):
23      str=str+"* "
24  print(str)
25  str=""
26
27
28  design=[1,2,3,4,5]
29  for i in design:
30  for j in range(i):
31      let=chr(96+i).upper()
32      str=str+" "+let
33  print(str)
34  str=""
35
```

Ex_1_6.py > ...

```
37
38     design=[1,2,3,4,5]
39     strw=""
40     for i in design:
41         for k in range(5-i):
42             strw=strw+" "
43             for j in range(i):
44
45
46                 strw=strw+f'{j+1}'
47             l=i
48             while(l>1):
49
50                 strw=strw+f'{l-1}'
51                 l=l-1
52
53
54             print(strw)
55             strw=""
```

OUTPUT:-

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Windows PowerShell

Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

```
PS C:\Users\Puru\Desktop\PRIYANSH\College\PYTHON> python -u "c:\Users\Puru\Desktop\PRIYANSH\College\PYTHON\Ex_1_6.py"
*****
***  
**  
*  
 *  
 * *  
* * *  
* * * *  
* * * * *  
A  
B B  
C C C  
D D D D  
E E E E E  
 1  
 121  
12321  
1234321  
123454321  
PS C:\Users\Puru\Desktop\PRIYANSH\College\PYTHON>
```