

Programming with Python (22616)

```

Python 3.7.1 Shell
File Edit Shell Debug Options Window Help
Python 3.7.1 (v3.7.1:260ec2c3d6a, Oct 20 2018, 14:05:18) [MSC v.1915 32 bit (I...
1|1 on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print('Hello Python')
Hello Python
>>>

```

VIII. Resources required

Sr. No.	Name of Resource	Specification	Quantity	Remarks (If any)
1.	Computer System	Computer (i3-i5 preferable RAM>2GB)	As per Batch Size	For ALL Experiments
2.	Operating System	Windows/Linux		
3.	Development Software	Pyhton IDE		

IX. Resources used (Additional)

Sr. No.	Name of Resource	Specification	Quantity	Remarks (If any)
1.	Computer System			
2.	Operating System			
3.	Development Software			

X. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Write steps for installing Python on window.
2. State IDLE in Python.
3. List key features of Python
4. Explain Python Path
5. State use of pep and pip

(Space for answers)

- i) Open any internet browser. Type <http://www.Python.org/downloads/> in address bar & Enter.
ii) Home page of Python will have displayed as.....

shown in Fig. 1
iii) Click on download the latest version for
Windows, which shows latest version as
shown in Fig. 2
iv) Open the Python 3.7.1 version pack and
double click on it to start installation and
installation windows will be open as shown
in Fig. 3.
v) Click on next install now for installation
and then setup progress windows will be
opened as shown in Fig. 4.
vi) After complete the installation, Click on
close button in the windows as shown in figs.

2) → IDLE (short for integrated Development
and Learning Environment) is an integrated
development environment for Python.

3) → Features of Python :-
• Easy to code.
• Free and open source.
• Object-oriented language.
• GUI Programming support.
• High Level Language.

4) → Python Path is an environment variable which
you can set to add additional directories
where python will look for modules and
packages.

XI. Exercise

- (Use blank space for answers or attach more pages if needed)
1. Print the version of Python
 2. Write steps to be followed to load Python interpreter in windows.

(Space for answers)

```
1) import platform  
print(platform.python_version())
```

2)

- 1. Click On your desktop, Click on the Start button, then click on Run.
2. On your desktop, click on the Start button, then click on Programs, Then Python 2.3.
3. In the third method of running the Python interpreter, you execute the interpreter by typing its file name at an MS DOS shell command line.



64MP AI QUAD CAMERA
Shot by yuvilove

2021/04/11 14:58

Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. List different modes of Programming in Python
2. Describe procedure to execute program using Interactive Mode
3. State the steps involved in executing the program using Script Mode
4. State the procedure to make file executable

(Space for answers)

Q) 1) Script
2) Interactive

2) The Interactive mode involves running your codes directly on the Python Shell which can be accessed from the terminal of the operating system. In the script mode, you have to create a file, give it a name with a .py extension, then runs your code. The interactive mode is suitable when running a few lines of code.

Q) 1. File Z New File or Press Ctrl+N

2. Write the Python code as function i.e. Script

3. Save it (`1.S`)

4. Execute it in interactive mode by using RUN option (`1.F5`)

4) 1. Open a terminal

2. Browse to the folder where the executable file is stored

3. Type the following command for any .bin file:
`Sudo chmod +x filename.bin` for any .run file.
`Sudo chmod +x filename.run`.

4. When asked for, type the required password and press Enter.



XI. Exercise

Note: Faculty must ensure that every group of students use different input value.

(Use blank space for answers or attach more pages if needed)

1. Write a Python program to display your name using Interactive Mode
2. Write a Python program to display "MSBTE" using Script Mode

(Space for answers)

1) → Print def personal_details():
 name:
 >>> print("Name:" + input("Enter the name"))

2) → print("MSBTE")

O/P:-

MSBTE

Programming with Python (22616)

	shift	left by the number of bits specified by the right operand.		
--	-------	------------------------------------------------------------	--	--

VIII. Resources required

Sr. No.	Name of Resource	Specification	Quantity	Remarks (If any)
1.	Computer System	Computer (i3-i5 preferable RAM>2GB)	As per Batch Size	For ALL Experiments
2.	Operating System	Windows/Linux		
3.	Development Software	Python IDE		

IX. Resources used (Additional)

Sr. No.	Name of Resource	Specification	Quantity	Remarks (If any)
1.	Computer System			
2.	Operating System			
3.	Development Software			

X. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Mention the use of //, **, % operator in Python
2. Describe ternary operator in Python
3. Describe about different Logical operators in Python with appropriate examples.
4. Describe about different Arithmetic operators in Python with appropriate examples.
5. Describe about different Bitwise operators in Python with appropriate examples.

(Space for answers)

1) → // :- Division of operands where the solution is a quotient left after removing
 ** :- Performs exponential calculation.
 % :- Returns the remainder after dividing the left operand with the right operand.

2) → Ternary operators also known as conditional expressions are operators that evaluate something based on a condition being true or



false. It was added to python in version 2.5 simply allows to test a condition.

3)

→ Logical operators perform Logical AND, Logical OR, and Logical NOT operations. For logical operators following condition are applied.

Operator	Meaning	Description	Example	Output
and	Logical AND	If both the operands are true then cond ⁿ becomes true	>>>(8>9) and (2<9) >>>(9>8) and (2<9)	False True
Or	Logical OR	If any of the two operands are non-zero then cond ⁿ becomes true	>>>(2>2) or (9<20) >>>(3!=3) or (9>20)	True False
not	Logical NOT	Used to reverse the logical state of its operand	>>>not(8>2) >>>not(2>10)	False True

5) → Bitwise operations acts on bits & performs bit by bit operation.

i) & [Bitwise AND]- Operator copies a bit, to the result, if it exists in both operands.

e.g. - >>>(a&b) = 0

ii) | [Bitwise OR]- It copies a bit, if it exists in either operand. e.g. ->>>(a|b) = 14

iii) ~ [Bitwise Not]- It is unary and has the effect of 'flipping' bits. e.g. ->>>(~a) = -11

[Space to Write Answers]

5)

→ Bitwise Operators:-

i) & [Bitwise AND] - operation copies a bit, to the result, if it exists in both operand.

eg:- $>>> (a \& b)$

ii) | [B]

4)

→ Arithmetic Operators:-

i) + [Addition] - Adds the value of the left and right operands.

eg:- $>>> 1 + 4$

ii) - [Subtraction] - Subtracts the value of the right operand from left operand

eg:- $>>> 10 - 5$

iii) * [Multiplication] - Multiplies the value of the left and right operand.

eg:- $>>> 5 * 2$

iv) / [Division] - Divides the value of the left operand by the right operand

eg:- $>>> 10 / 2$

v) ** [Exponent] - Performs exponential calculation.

eg:- $>>> 2 ** 3$

vi) % [Modulus] - Returns the remainder after dividing the left operand with the right operand.

eg:- $>>> 15 \% 4$

vii) // [Floor Division] - Division of operands where the solution is a quotient left after removing decimal numbers.



Exercise

*Note: Faculty must ensure that every group of students use different input value.
(Use blank space for answers or attach more pages if needed)*

1. Write a program to convert U.S. dollars to Indian rupees.
2. Write a program to convert bits to Megabytes, Gigabytes and Terabytes
3. Write a program to find the square root of a number
4. Write a program to find the area of Rectangle
5. Write a program to calculate area and perimeter of the square
6. Write a program to calculate surface volume and area of a cylinder.
7. Write a program to swap the value of two variables

(Space for answers)

1) → usd = float(input("Enter the currency in USD:"))
inr = usd * 73

print("The currency in INR is", round(inr, 2))

2)

→ def convert_bytes(bytes_number):
tags = ["Byte", "Megabyte", "Gigabyte", "Terabyte"]
i = 0
double_bytes = bytes_number

while (i < len(tags)) and bytes_number >= 1024:
double_bytes = bytes_number / 1024.0
i = i + 1
bytes_number = bytes_number / 1024

print(convert_bytes(4896587482345))
print(convert_bytes(9876524362))
print(convert_bytes(10248000))
print(convert_bytes(1048576))
print(convert_bytes(1024000))
print(convert_bytes(475445))
print(convert_bytes(1024))
print(convert_bytes(75))
print(convert_bytes(0))

English (22101)

*Exercise

4) → W = float(input('Please Enter the width of a rectangle:'))
h = float(input('Please Enter the height of a rectangle:'))

$$\text{Area} = \text{width} \times \text{height}$$

$$\text{perimeter} = 2 * (\text{width} + \text{height})$$

print("The Area of a rectangle is: %.2f" % Area)

5) ~~scint~~ ~~s = int~~ C input C "Side: " 10
area = s * s

$$\text{perimeter} = 4 * s$$

```
print("Area = " + str(area))
print("Perimeter = " + str(perimeter))
```

```

6)
→ Pi = 22/7
height = float(input("Height of cylinder"))
radian = float(input("Radius of cylinder"))
Volume = pi * radian * radian * height
Sur_area = ((2 * pi * radian) * height) +
           ((pi * radian ** 2) * 2)
print("Volume is:", Volume)
print("Surface Area is:", Sur_area)

```

$$\begin{aligned}x &= 10 \\y &= 50 \\ \text{temp} &= x \\ x &= y\end{aligned}$$

Maharashtra State Board of Technical Education

Print CII
S-temp

Print("Value of x:");
Print("Value of y:");
64MP AI QUAD CAMERA
Shot by yuvi ❤

```

3) #input a number
→ digit = int(input("Enter an integer
number:"))
#calculate square
square = digit * digit
print(f"Square of {digit} is {square}")

```

XI. References / Suggestions for further Reading

1. https://www.w3schools.com/Python/Python_operators.asp
2. <https://www.geeksforgeeks.org/basic-operators-Python/>
3. <https://www.guru99.com/Python-operators-complete-tutorial.html>
4. https://www.tutorialspoint.com/Python/Python_basic_operators
5. <https://www.youtube.com/watch?v=v5MR5JnKcZI>
6. https://www.youtube.com/watch?v=fN8yDMdI_aw

XII. Assessment Scheme

Performance indicators		Weightage
Process related (35 Marks)		70%
1	Logic Formulation	10%
2	Debugging Ability	20%
3	Follow ethical practices	40%
Product related (15 Marks)		30%
4	Expected output	10%
5	Timely Submission of report	10%
6	Answer to sample questions	10%
Total (50 Marks)		100%

List of student Team Members

1. Shreyashi Sumwase
2. Dipashri Kokate
3. Sanu Deskar
4.

Marks Obtained			Dated signature of Teacher
Process Related (35)	Product Related (15)	Total (50)	



VIII. Resources required

Sr. No.	Name of Resource	Specification	Quantity	Remarks (If any)
1.	Computer System	Computer (i3-15 preferable RAM>2GB)	As per Batch Size	For ALL Experiments
2.	Operating System	Windows/Linux		
3.	Development Software	Python IDE		

IX. Resources used (Additional)

Sr. No.	Name of Resource	Specification	Quantity	Remarks (If any)
1.	Computer System			
2.	Operating System			
3.	Development Software			

X. Practical related Questions

Note: Below given are few sample questions for reference, Teachers must design more such questions to ensure the achievement of identified CO.

1. List operators used in if conditional statement
2. Differentiate between if-else and nested-if statement

(Space for answers)

- 1) If-else Statement
i) Nested-if Statement
- 2) If-else Statement - The if statement alone tells us that if a condition is true it will execute a block of statements and if the condition is false it won't. But what if we want to do something else if the condition is false. Here comes the else statement.
ii) Nested-if Statement - A nested if is another statement that is the target of another if statement. Nested if statement means an if statement inside another if statement.



64MP AI QUAD CAMERA
Shot by yuvi

2021/04/11 15:07

XI. Exercise*Note: Faculty must ensure that every group of students use different input value.*

(Use blank space for answers or attach more pages if needed)

1. Write a program to check whether a number is even or odd
2. Write a program to find out absolute value of an input number
3. Write a program to check the largest number among the three numbers
4. Write a program to check if the input year is a leap year or not
5. Write a program to check if a Number is Positive, Negative or Zero
6. Write a program that takes the marks of 5 subjects and displays the grade.

(Space for answers)

```
1)
num = int(input("Enter a number:"))
if num % 2 == 0:
    print("It is Even", format(num))
else:
    print("It is Odd", format(num))
```

2) number
complex_num = (3+1j)

print("The magnitude of the complex
number is .", abs(complex_num))

3) num1=10
num2=14
num3=12

```
if (num1 >= num2) and (num1 >= num3):
    largest = num1
elif (num2 >= num1) and (num2 >= num3):
    largest = num2
else:
    largest = num3

print("The largest number is", largest)
```



4)
→ year = int(input("Enter a year."))
if (year % 4) == 0:
 if (year % 100) == 0:
 if (year % 400) == 0:
 print("203 is a leap year.", format(year))
 else:
 print("203 is not a leap year.", format(year))
 else:
 print("203 is a leap year.", format(year))
else:
 print("203 is not a leap year.", format(year))

5)
→ num = float(input("Enter a number."))
if num > 0:
 print("positive number")
elif num == 0:
 print("Zero")
else:
 print("Negative number")

6)
→ sub1 = int(input("Enter marks of first subject."))
sub2 = int(input("Enter marks of second subject."))
sub3 = int(input("Enter marks of third subject."))
sub4 = int(input("Enter marks of fourth subject."))
sub5 = int(input("Enter marks of fifth subject."))
avg = (sub1 + sub2 + sub3 + sub4 + sub5)/5
if (avg >= 90):
 print("Grade: A")
elif (avg >= 80 & avg < 90):
 print("Grade: B")
elif (avg >= 70 & avg < 80):
 print("Grade: C")
else:
 print("Grade: D")

Maharashtra State Board of Technical Education

64MP AI QUAD CAMERA
Shot by yuvi ❤️

D
→ print X,
Syntax error: Missing parentheses in call to print
2)
→ print X,
A ^ SyntaxError: Missing parentheses in call to
print



64MP AI QUAD CAMERA
Shot by yuv/

2021/04/11 15:02

XI. Exercise

Note: Faculty must ensure that every group of students use different input value.

(Use blank space for answers or attach more pages if needed)

1. Print the following patterns using loop:

a. *

**

*

b. *

**

*

c. 1010101

10101

101

1

2. Write a Python program to print all even numbers between 1 to 100 using while loop.
3. Write a Python program to find the sum of first 10 natural numbers using for loop.
4. Write a Python program to print Fibonacci series.
5. Write a Python program to calculate factorial of a number
6. Write a Python Program to Reverse a Given Number
7. Write a Python program takes in a number and finds the sum of digits in a number.
8. Write a Python program that takes a number and checks whether it is a palindrome or not.

(Space for answers)

1)

```
def pyprint(n):  
    for i in range(0,n):  
        for j in range(0,i+1):  
            print("*",end="")  
        print()  
    print("n")  
    pyprint(n)
```

```
2>#input("Enter the no. of rows: ")  
#K=2  
for i in range(0,K):  
    for j in range(0,i+1):  
        print("*",end="")  
    print()  
for i in range(K+1,K+2):  
    for j in range(0,K+1-i):  
        print("*",end="")  
    print()
```



```

# inner loop is used to print no. of space
for j in range(0, k):
    print(" " * j)
# Decrement in k after each iteration
k = k - 1
# This inner loop is used to print stars
for j in range(0, i + 1):
    print("*" * j, end=" ")
print("\n")
# Downward triangle pyramid
k = rows - 2 # used to print the space
for i in range(rows, -1, -1):
    for j in range(k, 0, -1):
        print(end=" ")
    k = k + 1
    for j in range(0, i + 1):
        print("*" * j, end=" ")
    print("\n")

```

c) rows = int(input("Enter the no. of rows:"))

```

k = 2 * rows - 2
for i in range(rows, -1, -1):
    for j in range(k, 0, -1):
        print(end=" ")
    k = k + 1
    for j in range(0, i + 1):
        print("*" * j)
    print("\n")

```

2) i = 0

```

while i <= 100:
    if (i % 2 == 0):
        print("%d" % i)
    i = i + 1

```

[Space to Write Answers]

3)

```

→ i = n = int(input("Enter a number:"))
Sum1 = 0
for n in range(0, n):
    Sum1 = Sum1 + n
print("The sum of first 10 natural
numbers from 1 to 103 = " + str(Sum1))

```

4)

```

→ n = int(input("Enter the value of n:"))
a = 0
b = 1
sum = 0
count = 1
print("Fibonacci series:", end = " ")
while count <= n:
    print(sum, end = " ")
    count += 1
    a = b
    b = sum
    sum = a + b

```

5)

```

→ num = 7
factorial = 1
if num < 0:
    print("Sorry, factorial does not
exist for negative numbers")
elif num == 0:
    print("The factorial of 0 is 1")
else:
    for i in range(1, num + 1):

```

PrintC "The number is not a palindrome"

else:

PrintC "The number is a palindrome!!"

If temp == rev

temp = 0

rev = rev * 10 + digit

digit = n % 10

while (n > 0):

n = int(input("Enter Number:"))

PrintC "The total sum of digits is", sum

PrintC "The total sum of digits is", sum

total = 0

digit = n % 10

total = total + digit

while (n > 0):

n = int(input("Enter a number:"))

PrintC "Reverse of the number:", rev

rev = 0

rev = rev * 10 + digit

digit = n % 10

while (n > 0):

n = int(input("Enter a number:"))

PrintC "The factorial of", num, "is", factomial

factomial = 1

for i in range(1, num+1):

factomial *= i

Programming v

x. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. When to used list
2. Describe various list functions
3. Write syntax for a method to count occurrences of a list item in Python
4. How to concatenate list
5. Justify the statement "lists are mutable"
6. Describe the use pop operator in list

(Space for answers)

i) They are used to store an ordered collection of items, which might be of different types but usually they aren't

2) copy() - Returns a copy of the list
3) clear() - Removes all the elements from list
4) append() - Adds an element at the specified position
5) insert() - Adds an element at the specified position
6) reverse() - Reverses the order of the list
7) sort() - Sorts the list

3) Syntax - `sort(key=..., reverse=...)`

4) Syntax - `list_name.count(object)`

5) Using Native Method . Using `textools.chano`
• Using + operator
• Using list comprehension
• Using extend()
• Using * Operator

Q)

→ Lists are
the order
an item

→ pop()
that re
from th
value. T
and re
given, th
but and

A)

2021/04/11 15:04



64MP AI QUAD CAMERA
Shot by yuvi

© Maharashtra State Board of Technical Education

Practical Related Questions:-

Q)

→ Lists are mutable because you can change the order of items in a list or reassign an item in a list.

Q)

→ `pop()` is an inbuilt function in Python that removes and returns last value from the list or the given index value. The value at index is popped out and removed. If the index is not given, then the last element is popped out and removed.

XI. Exercise

Note: Faculty must ensure that every group of students use different input value.

- (Use blank space for answers or attach more pages if needed)
1. Write a Python program to sum all the items in a list.
 2. Write a Python program to get the largest number from a list.
 3. Write a Python program to multiply all the items in a list.
 4. Write a Python program to reverse a list.
 5. Write a Python program to get the smallest number from a list.
 6. Write a Python program to find common items from two lists.
 7. Write a Python program to select the even items of a list.

(Space for answers)

```

1) → list[]

num = int(input("How many numbers?"))
for n in range(num):
    numbers = int(input("Enter number"))
    list.append(numbers)
print("Sum of elements in given list is:",sum(list))

2)

def multiply(list_items):
    tot = 1
    for x in items:
        tot *= x
    return tot
print(multiply([1, 2, -8]))

```

```

3) list1 = [10, 20, 4, 45, 99]
list1.sort()
print("Largest element is:", list1[-1])

```



4) Exercise [Space to Write Answers]

\rightarrow def smallest_num_in_list(list):
 min = list[0]
 for a in list:
 if a < min:
 min = a
 return min.

\rightarrow print(smallest_num_in_list([1, 2, -8, 0]))
 \rightarrow def reverse(list):
 return [ele for ele in reversed(list)]
 list = [10, 11, 12, 13, 14, 15]
 print(reverse(list))

5)
 \rightarrow list1 = [1, 2]
 list2 = [2, 3]

list1 = set(list1)
 list2 = set(list2)
 intersection = list1 - list2 - set.intersection(list1, list2)
 intersection = list1.intersection(list2)
 print(intersection)

6)
 \rightarrow def sum_of_numbers(list1):
 tot = 0
 for i in list1:
 tot += i
 return tot
 print(sum_of_numbers([1, 2, 3, 4, 5, 6, 7, 8, 9, 10]))

7)
 \rightarrow list1 = [1, 2, 3, 4, 5, 2, 3, 6, 4, 2, 1, 2]
 for num in list1:
 if num % 2 == 0:
 print("num", end = " ")

\Rightarrow Syntactic - empty tuple
2) \Rightarrow Syntactic - empty tuple

\Rightarrow empty tuple = tuple

Syntax - empty tuple = .
use static type.
Sequence of Python objects which
structure. Tuple use to store
supports a new tuple as a distinct
empty brackets is like list Python

(Space for answers)

- Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.
1. Define empty tuple. Write syntax to create empty tuple.
 2. Write syntax to copy specific elements existing tuple into new tuple.
 3. Compare tuple with list (Any 4 points)

X.

Practical related Questions

Sr.	Name of Resource	Specification	Quantity	Remarks (If any)
1.	Computer System	Computer (i3-i5 preferable RAM>2GB)	As per Batch	
2.	Operating System	Windows/Linux	For ALL	Experiments
3.	Development Environment	Python IDE		Software

IX. Resources used (Additional)

Sr.	Name of Resource	Specification	Quantity	Remarks (If any)
1.	Computer System	Computer (i3-i5 preferable RAM>2GB)	As per Batch	
2.	Operating System	Windows/Linux	For ALL	Experiments
3.	Development Environment	Python IDE		Software

VIII. Resources required

Programming with Python (22616)

2021/04/11 15:04



64MP AI QUAD CAMERA
Shot by yuvih

- IV) The implementation of Tuples is time-consuming
and it takes a lot of time to iterate through tuples.
V) Unexpected exceptions can occur in tuples.
VI) It is difficult to change elements in tuples.

i) Does not have many built-in methods
Many built-in methods are available

ii) Consumes more memory
Same operations take more memory

iii) Consumes more memory
Same operations take more memory

iv) No decorations can work on lists
Decorations can work on tuples

v) A tuple is immutable A list is mutable

3) ↪ Tuples Lists

XI. Exercise

Note: Faculty must ensure that every group of students use different input value.

(Use blank space for answers or attach more pages if needed)

1. Create a tuple and find the minimum and maximum number from it.
2. Write a Python program to find the repeated items of a tuple.
3. Print the number, in words, for Example: 1234 => One Two Three Four

(Space for answers)

1) $t = (1, 2, 3)$
minimum = min(t)
print(minimum)
maximum = max(t)
print(maximum)

2) $tuplex = 2, 4, 5, 6, 2, 3, 4, 5, 7$
print(tuplex)
count = tuplex.count(4)
print(count)

3) num = input("Enter a number : ")
numTup = tuple(num)
numWords = ["One", "Two", "Three", "Four"]
if num in numTup:
 print(numWords[int(num)])

Programming with Python**VIII. Resources required**

Sr. No.	Name of Resource	
1.	Computer System	Comp. preferable
2.	Operating System	Windows/Linux
3.	Development Software	Python IDE

IX. Resources used (Additional)

Sr. No.	Name of Resource	Specification	Quantity	Remarks (If any)
1.	Computer System			
2.	Operating System			
3.	Development Software			

X. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Describe the various set operations
2. Describe the various methods of set
(Space for answers)

1) The sets in python are typically used for mathematical operations like union, intersection, difference, and complement etc.

2) a) Creating a set - A set is created by using set() function or placing all the elements within a pair of curly braces.

b) Accessing values in set - We cannot access individuals values in set. We can only access all elements together.

c) Updating items in a set - We can add element to a set by using add() method.

d) Removing items in set - We can remove elements from a set by using discard() method.

50



XI. Exercise

Note: Faculty must ensure that every group of students use different input value.

(Use blank space for answers or attach more pages if needed)

1. Write a Python program to create a set, add member(s) in a set and remove one item from set.
2. Write a Python program to perform following operations on set: intersection of sets, union of sets, set difference, symmetric difference, clear a set.
3. Write a Python program to find maximum and the minimum value in a set.
4. Write a Python program to find the length of a set.

(Space for answers)

1)

```
→ Color_set = set()
color_set.add("Red")
print(color_set)
color_set.update(["Blue", "Green"])
print(color_set)
color_set.remove("Blue")
print(color_set)
```

2)

```
→ A = {1, 2, 4, 6, 8}
B = {1, 2, 3, 4, 5}
print("Union:", A|B)
print("Intersection:", A&B)
print("Difference:", A-B)
print("Symmetric Difference:", A^B)
print("Clear sets:", )
```

3)

```
→ seta = set([5, 10, 3, 15, 2, 20])
print(max(seta))
print(min(seta))
```



Programming with Python (22616)

4)
→ Set a = set([5,10,3,15,2,20])
print(len(seta))



64MP AI QUAD CAMERA
Shot by yuvilove

2021/04/11 15:05

XI. Exercise *Faculty must ensure that every group of students use different input value.*

Note: Faculty must ensure that every group of students use different input value.
(Use blank space for answers or attach more pages if needed)

1. Write a Python script to sort (ascending and descending) a dictionary by value.
2. Write a Python script to concatenate following dictionaries to create a new one.
3. Sample Dictionary:
 - a. dic1 = {1:10, 2:20}
 - b. dic2 = {3:30, 4:40}
 - c. dic3 = {5:50,6:60}
4. Write a Python program to combine two dictionary adding values for common keys.

```
keys.
d1 = {'a': 100, 'b': 200, 'c':300}
d2 = {'a': 300, 'b': 200,'d':400}
# Print original dictionaries.
print("Original dictionaries:")
print(d1)
print(d2)
# Sort the dictionary by value.
sorted_d = sorted(d.items(), key = lambda item: item[1], reverse=True)
# Print sorted dictionary.
print("Sorted dictionary by value:")
print(sorted_d)
```
5. Write a Python program to find the highest 3 values in a dictionary.

```
Sample Data: [{"V": "S001"}, {"V": "S002"}, {"V": "S001"}, {"V": "S005"}, {"V": "S005"}, {"V": "S009"}, {"V": "S007"}]
```

(Space for answers)

```
1) import operator
d = {1: 2, 2: 3, 3: 4, 4: 1, 5: 0, 6: 0}
original_dictionary = d
print("Original dictionary:")
print(d)
sorted_d = sorted(d.items(), key = operator.itemgetter(1), reverse=True)
key = operator.itemgetter(0)
print("Dictionary in ascending order:")
print(sorted_d)
by_value = sorted(d.items(), key = operator.itemgetter(1), reverse=False)
sorted_d = dict(by_value)
key = operator.itemgetter(0)
print("Dictionary in descending order:")
print(sorted_d)
```

2021/04/11 15:06

```
2) dic1 = {1: 10, 2: 20}
dic2 = {3: 30, 4: 40}
```

```
dic3 = { 'a':50, 'b':60 }
dic4 = { 'c':100, 'd':200, 'e':300 }
for d in dic4, dic3, dic2, dic1:
    update(d)
print(dic4)
```

```
2) d1 = { 'a':100, 'b':200, 'c':300 }
d2 = { 'a':200, 'b':300, 'd':400 }
d3 = { 'a':300, 'b':400, 'c':500 }
for i in d1.items():
    for j in d2.items():
        if i[0] == j[0]:
            d1[i[0]] = i[1] + j[1]
my_results = { 'a':400, 'b':600 }
```

```
3) l = [ { 'V1':500, 'V2':1000, 'V3':1500 },
        { 'V1':500, 'V2':1000, 'V3':1500 } ]
print("Original list : ", l)
print("Dictionary with 3 highest values")
high_k = {}
for i in l:
    if len(high_k) < 3:
        high_k[i['V1']] = i['V2']
    else:
        maxv = max(high_k.values())
        minv = min(high_k.values())
        if maxv < i['V2']:
            high_k.pop(minv)
            high_k[i['V1']] = i['V2']
print("Dictionary with unique values : ", high_k)
print("Count : ", len(high_k))
```

```
4) from collections import Counter
my_dict = { 'T':2, 'A':3, 'G':2, 'C':1 }
my_counter = Counter(my_dict)
k = my_counter.most_common(3)
highest_values = k[-1][1]
print("Dictionary with 3 highest Values : ", highest_values)
print("Keys : ", my_counter.keys())
print("Count : ", my_counter)
for i in my_counter:
    print(i, " : ", my_counter[i])
print("Count : ", len(my_counter))

5) print("C10 : ", C10)
print("C11 : ", C11)
print("C12 : ", C12)
print("C13 : ", C13)
print("C14 : ", C14)
print("C15 : ", C15)
print("C16 : ", C16)
print("C17 : ", C17)
print("C18 : ", C18)
print("C19 : ", C19)
print("C20 : ", C20)
```

strip()	Remove spaces at the beginning and at the end of the string.
startswith()	Returns true if the string starts with the specified value
title()	Converts the first character of each word to upper case
translate()	Returns a translated string
upper()	Converts a string into upper case

VIII. Resources required

Sr. No.	Name of Resource	Specification	Quantity	Remarks (If any)
1.	Computer System	Computer (i3:45 preferable RAM >2GB)	As per Batch Size	For All Experiments
2.	Operating System	Windows/Linux		
3.	Development Software	Python IDE		

Sr. No.	Name of Resource	Specification	Quantity	Remarks (If any)
1.	Computer System			
2.	Operating System			
3.	Development Software			

X. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. Describe about string formatting operator with example.
2. Give the syntax and example of title() and capitalize() methods.
3. Give the syntax and significance of string functions: title() and strip().
(Space for answers)

→ Python uses f-Style String. The f-Style creates new, formatted strings. The f-Style is used to format a set of variables enclosed in a "whole" f together with a format string which contains normal text together.

2) → str.title() is string example.
eg: str("this is string example").title()
.....

```
str.capitalize()  
eg: str = "this is string example... wwww!!!"  
print(str.capitalize())
```

- 3) ~~String.strip()~~ - Syntax
Significance - returns a copy of the string with remove all the leading & trailing spaces from a string.
~~String.title()~~ Significance - returns a string with first letter of each word capitalized, a title cased string.

XI. Exercise

Note: Faculty must ensure that every group of students use different input value.

(Use blank space for answers or attach more pages if needed)

1. Write a Python function that accepts a string and calculate the number of upper case letters and lower case letters.
2. Write a Python program to generate a random float where the value is between 5 and 50 using Python math module.

(Space for answers)

```
1) def string_test(s):  
    count_upper_case = 0  
    count_lower_case = 0  
    for c in s:  
        if c.isupper():  
            count_upper_case += 1  
        elif c.islower():  
            count_lower_case += 1  
        else:  
            pass  
    print("Original String : ", s)  
    print("No. of Upper Case characters: ", count_upper_case)  
    print("No. of Lower Case characters: ", count_lower_case)
```

```
print("No. of Lower case characters:",  
      len([c for c in s if c.islower()]))
```

```
String-test('The quick Brown Fox')
```

2)

```
→ import random  
for x in range(5):  
    print("%0.3f" % random.  
         uniform(0, 50)), end =',')
```



64MP AI QUAD CAMERA
Shot by yuvicloud

2021/04/11 15:06

IX. Resources used (Additional)

Sr. No.	Name of Resource	Specification	Quantity	Remarks (If any)
1.	Computer System			
2.	Operating System			
3.	Development Software			

X. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. What is the output of the following program?

def myfunc(text, num):

 while num > 0:

 print(text)

 num = num - 1

 myfunc('Hello', 4)

2. What is the output of the following program?

num = 1

def func0:

 num = 3

 print(num)

func0

 print(num)

(Space for answers)

1) *No output*

2) *No output*

Exercise

Note: Faculty must ensure that every group of students use different input value.

(Use blank space for answers or attach more pages if needed)

1. Write a Python function that takes a number as a parameter and check the number is prime or not.
2. Write a Python function to calculate the factorial of a number (a non-negative integer). The function accepts the number as an argument.
3. Write a Python function that accepts a string and calculate the number of upper case letters and lower case letters.

(Space for answers)

```
1) def prime(num):  
    if (num == 1):  
        return False  
    elif (num == 2):  
        return True  
    else:  
        for x in range(2, num):  
            if (num % x == 0):  
                return False  
        return True  
print(prime(16))  
  
2) def factorial(n):  
    if n == 0:  
        return 1  
    else:  
        return n * factorial(n-1)  
n = int(input("Input a number to  
compute the factorial:"))  
print(factorial(n))
```



```
3) →def string_test(s):
    d = {"UPPER CASE":0, "LOWER CASE":0}
    for c in s:
        if c.isupper():
            d["UPPER CASE"]+=1
        elif c.islower():
            d["LOWER CASE"]+=1
        else:
            pass
    print("Original String : ", s)
    print("No. of Upper Case Character :",
          d["UPPER CASE"])
    print("No. of Lower Case Characters :",
          d["LOWER CASE"])
    string_test("The quick Brown Fox")
```



(Space for answers)

→ print math.sqrt(25)
SyntaxError: invalid syntax

2) → print random.randint(0,5)
SyntaxError: invalid syntax

3) → print time.time()
SyntaxError: invalid syntax

XI. Exercise

Note: Faculty must ensure that every group of students use different input value.

(Use blank space for answers or attach more pages if needed)

1. Write a Python program to create a user defined module that will ask your college name and will display the name of the college.
2. Write a Python program that will calculate area and circumference of circle using



64MP AI QUAD CAMERA
Shot by yuvi ❤️

2021/04/11 15:07

(Space for answers)

```
1) def college_name():
    name = input("Enter your College name:")
    print("Brahmadeshada Mane Polytechnic", name)
```

```
2) import math
area = math.pi * radius ** 2
area = math.pi * radius * 2
circumference = 2 * math.pi * radius
print("Area = 10.4 ft. ^2", area)
print("Circumference = 10.4 ft.", circumference)
```

```
3) import calendar
yy = int(input("Enter year:"))
mm = int(input("Enter month:"))
print(calendar.month(yy, mm))
```

i)

(Space for answers)

→ NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on array are performed.

2)

→ The programming tools are used which suit problem and environment. It is open source and does not cost anything and use general-purpose language.

XI. Exercise

Note: Faculty must ensure that every group of students use different input value.

(Use blank space for answers or attach more pages if needed)

1. Write a Python program to create two matrices and perform addition, subtraction, multiplication and division operation on matrix.
2. Write a Python program to concatenate two strings.
3. Write a NumPy program to generate six random integers between 10 and 30.

(Space for answers)

2)

→ str1 = "Hello"
str2 = "Devansh"
Str3 = str1 + str2
print(Str3)

83

3)

```
→ import numpy as np
x=np.random.randint(low=10, high=30,
size=6)
print(x)
```

4)

```
→ #Addition of two matrices
```

```
M1 = [[1, 2], [4, 1]]
M2 = [[3, 2], [1, 2]]
add = [[0, 0], [0, 0]]
for i in range(len(M1)):
    for j in
range(len(M1[0])):
        add[i][j] = M1[i]
[j] + M2[i][j]
for num in add:
    print(num)
```

```
#Subtraction of two matrices
```

```
M1 = [[1, 2], [4, 1]]
M2 = [[3, 2], [1, 2]]
sub = [[0, 0], [0, 0]]
for i in range(len(M1)):
    for j in range(len(M1[0])):
        add[i][j] = M1[i][j] - M2[i][j]
for num in sub:
    print(num)
```



[Space to Write Answers]

#Multiplication of two matrices

 $M1 = [[1, 2], [4, 1]]$ $M2 = [[3, 2], [1, 2]]$ $mul = [[0, 0], [0, 0]]$

for i in range(len(M1)):

for j in range(len(M1[0])):

mul[i][j] = M1[i][j] * M2[i][j]

for num in mul:

print(num)

#Division of two matrices

 $M1 = [[1, 2], [4, 1]]$ $M2 = [[1, 2], [1, 2]]$ $div = [[0, 0], [0, 0]]$

for i in range(len(M1)):

for j in range(len(M1[0])):

div[i][j] = M1[i][j] / M2[i][j]

for num in div:

print(num)

(Space for answers)

- 1) Method Overriding - In Python, method overriding means creating two methods with the same name but different implementations.
- Method Overloading - In Python, method Overloading means we need to write the method logic in such a way that depending upon the parameters, different piece of code executes inside the function.

- 2) Traceback (most recent call last):
File "<string>" line 1 in <module>
NameError: name 'Animal' is not defined

XI. Exercise

Note: Faculty must ensure that every group of students use different input value.

(Use blank space for answers or attach more pages if needed)

1. Write a Python program to create a class to print an integer and a character with two methods having the same name but different sequence of the integer and the character parameters. For example, if the parameters of the first method are of the form (int n, char c), then that of the second method will be of the form (char c, int n)
2. Write a Python program to create a class to print the area of a square and a rectangle. The class has two methods with the same name but different number of parameters. The method for printing area of rectangle has two parameters which are length and breadth respectively while the other method for printing area of square has one parameter which is side of square.
3. Write a Python program to create a class 'Degree' having a method 'getDegree' that prints "I got a degree". It has two subclasses namely 'Undergraduate' and 'Postgraduate' each having a method with the same name that prints "I am an Undergraduate" and "I am a Postgraduate" respectively. Call the method by creating an object of each of the three classes.

(Space for answers)

Inheritance allows us to define a class that inherits all the methods and properties from another class. Parent class is the class being inherited from, also called base class. Child class is the class that inherits from another class, also called derived class.

2)

- Single inheritance
- Multilevel inheritance
- Multiple inheritance
- Multipath inheritance
- Hierarchical inheritance
- Hybrid inheritance

XI. Exercise

Note: Faculty must ensure that every group of students use different input value.

(Use blank space for answers or attach more pages if needed)

1. Create a class Employee with data members: name, department and salary. Create suitable methods for reading and printing employee information
2. Python program to read and print students information using two classes using simple inheritance.
3. Write a Python program to implement multiple inheritance

(Space for answers)

1) Class Employee:
def __init__(self, name, self, salary, department):
 self.name = name
 self.salary = salary
 self.department = department

2021/04/11 15:08

```
def showData(self):
    print("Name : " + self.name)
    print("Salary : " + self.salary)
    print("Department : " + self.department)

def main():
    emp = employee()
    emp.setData("Rajesh", "00,00,000", "Civil
    Enginner")
    emp.showData()

if name == "main":
    main()

2)
→ class Student:
    def __init__(self):
        self.name = input("Enter your name:")
        self.college = input("Enter your college
        name:")
        self.roll = int(input("Enter your
        roll number:"))

class Test(Student):
    def display(self):
        print("Student info is:")
        print("Name is : " + self.name)
        print("College Name is : " + self.college)
        print("Roll number is : " + self.roll)

obj = Test()
obj.display()
```

[Space to Write Answers]

3) class class1:
def m(self):
print("In class 1")

class class2(class1):
def m(self):
print("In class 2")

class class3(class1):
def m(self):
print("In class 3")

class class4(class2, class3):
pass

obj = class4()
obj.m

IX. Resources used (Additional)

Sr. No.	Name of Resource	Specification	Quantity	Remarks (If any)
1.	Computer System			
2.	Operating System			
3.	Development Software			

X. Practical related Questions

Note: Below given are few sample questions for reference. Teachers must design more such questions to ensure the achievement of identified CO.

1. State Exception.
2. How to handle exception in Python?

(Space for answers)

1)

→ Exception are uncovered errors that can occur during run time of a program.

2)

→ If the python program contains suspicious code that may throw the exception, we must place that code in the try block. The try block must be followed by with the except statement which contains a block of code that will be executed if there is some exception in the try block.

XI. Exercise

Note: Faculty must ensure that every group of students use different input value.

(Use blank space for answers or attach more pages if needed)

1. Write a Python program to Check for ZeroDivisionError Exception.
2. Write a Python program to create user defined exception that will check whether the password is correct or not?

(Space for answers)

1)

```
a = int(input("Enter a:"))
a = int(input("Enter b:"))
c = a/b
print("a/b = " + str(c))
except Exception:
    print("Can't divide by zero")
    print(Exception)
else:
    print("Hi I am else block!")
```

2)

```
import getpass
def userLogin(userName, password):
    if userName == 'Kwikedemy' and password == "Kwikedemy":
        return True
    else:
        raise Exception("User name & password not matched")
    userName = input("Enter user name:")
    password = getpass.getpass("Enter password:")
    result = userLogin(userName, password)
    if result == True:
        print("Logged in successfully!")
    else:
        print("User Name or Password is wrong")
```