Basic Questions and Programs

1. What is Python.
2. What is Programming language.
3. What is software and how many types.
4. What is python script.
5. What kind of applications we can develop using python.
6. How to develop standalone application in python.
7. How to develop distributed applications in python
8. Is python a case sensitive language
9. where we can use python.
10. WAP to Display Welcome To Sathya Technologies
11. WAP to accept the Student Name and Display a message Welcome Student Name
12. WAP to accept two numbers and perform addition, Subtraction, Multiplication and Division
13. WAP to find the type Of a Variable
14. WAP to accept student first name and last name and Display the student full Name
15. WAP to accept a str value and convert the string value to int and print int value.
16. WAP to accept a double value and convert the double value to str and print string value

17. WAP to accept a str value and convert the str value to double and print double value.
18. WAP to accept student no, student name, marks1, marks2, marks3 from command prompt and calculate total marks, average marks and print.
19. WAP to input two numbers and swap those values.
20. WAP to input Employee id_no, name, Basic salary, HRA, DA and Find out the Gross Salary.

Questions on Operators and Programs

1. How many types of operators in python.
2. What is the difference between logical and 'and' bitwise and.
3. What is the difference between logical or and bitwise or.
4. What are identity Operators.
5. Output of print(9//2)
6. What is the type a where a = 1,00,000
7. What is the type of 'inf'?
   a) Boolean
   b) Integer
   c) Float
   d) Complex
8. What does ~~~~~~5 evaluate to?
   a) +5
   b) -11
   c) +11
   d) -5
9. What is the purpose of not in operator.
10. What is the purpose pass statement in python.
11. Which function overloads the >> operator.

12. Output of format(10/3) and what is the return type.
13. Output of "Sathya" > "sathya".
14. Output of 10 and 10.
15. Output of 10 and 0.
16. Output of 0 and 10.
17. Output of 0 and 0.
18. Output of 0 or 10.
19. Which is the correct operator for power(x^y)?
    a) X^y
    b) X**y
    c) X^^y
    d) None of the mentioned
20. Which one of these is floor division?
    a) /
    b) //
    c) %
    d) None of the mentioned
21. What is the order of precedence in python?
    i) Parentheses ii) Exponential iii) Multiplication iv) Division
    v) Addition vi) Subtraction
    a) i,ii,iii,iv,v,vi
    b) ii,i,iii,iv,v,vi
    c) i,ii,iv,iii,v,vi
    d) i,ii,iii,iv,vi,v
22. Mathematical operations can be performed on a string. State whether true or false.
    a) True
    b) False
23. Operators with the same precedence are evaluated in which manner?
    a) Left to Right

    b) Right to Left
    c) Can't say
24. What is the output of this expression, 3*1**3?
    a) 27
    b) 9
    c) 3
    d) 1
25. Which one of the following have the same precedence?
    a) Addition and Subtraction
    b) Multiplication and Division
    c) Both Addition and Subtraction AND Multiplication and Division
    d) None of the mentioned
26. The expression Int(x) implies that the variable x is converted to integer. State whether true or false.
    a) True
    b) False
27. Which one of the following have the highest precedence in the expression?
    a) Exponential
    b) Addition
    c) Multiplication
    d) Parentheses
28. Which of the following is invalid?
    a) _a = 1
    b) __a = 1
    c) __str__ = 1
    d) none of the mentioned
29. Which of the following is an invalid variable?
    a) my_string_1
    b) 1st_string

c) foo
d) _

30. Which of the following is not a keyword?
a) eval
b) assert
c) nonlocal
d) pass

31. Which of the following is an invalid statement?
a) abc = 1,000,000
b) a b c = 1000 2000 3000
c) a,b,c = 1000, 2000, 3000
d) a_b_c = 1,000,000

32. Which of the following cannot be a variable?
a) __init__
b) in
c) it
d) on

33. What Are The Two Built-In Functions To Read A Line Of Text From Standard Input, Which Is By Default The Keyboard?

34. What Will Be The Output Of The Following Code Snippet?
```
txt = input("Enter your input: ")
print ("Received input is : ", txt)
```

35. What Will Be The Output Of The Following Code Snippet?
```
txt = eval(input("Enter your input: "))
print ("Received input is : ", txt)
```

36. Which Of The Following Are The Attributes Related To A File Object?
A. closed
B. mode
C. name
D. rename

37. Which Of The Following Statements Correctly Explain The Function Of Tell() Method?
A. tells the current position within the file.
B. indicates that the next read or write will occur at that many bytes from the beginning of the file.
C. move the current file position to a different location.
D. it changes the file position only if allowed to do so else returns an error.

38. Which Of The Following Statements Correctly Explain The Function Of Seek() Method?
A. tell the current position within the file.
B. indicate that the next read or write occurs from that position in a file.
C. determine if you can move the file position or not.
D. move the current file position to a different location at a defined offset.

Questions on Flow Control

(if, if-else, if-elif-else, nested-if, for and while)

1. WAP input two numbers and find out the biggest number.
2. WAP input three numbers and find out the biggest number.
3. WAP input any number and check it is positive or negative.
4. Given an int n, return the absolute difference between n and 21, except return double the absolute difference if n is over 21.

diff21(19) → 2
diff21(10) → 11
diff21(21) → 0

5. WAP input any positive number and check it is even or odd
6. Write a program input customer name and slab type(i/c/r) calculate units consumed
   Conditions: If slab type is industry then unit rate is 5.25/-
   If slab type is commercial then unit rate is 4.00/-
   If slab type is residence then unit rate is 2.08/-
   Calculate total bill.
7. Write a program input Employee name, salary, Designation(m/a/c) .
   If designation is manager then bonus is 20% on his salary
   If designation is analyst then bonus is 10% on his salary
   If designation is clerk then bonus is 5% on his salary.
   Calculate Total salary.
8. Given 2 ints, a and b, return True if one if them is 10 or if their sum is 10.
   makes10(9, 10) → True
   makes10(9, 9) → False
   makes10(1, 9) → True
9. Given 2 int values, return True if one is negative and one is positive. Except if the parameter "negative" is True, then return True only if both are negative
   pos_neg(1, -1, False) → True
   pos_neg(-1, 1, False) → True
   pos_neg(-4, -5, True) → True
10. We have a loud talking parrot. The "hour" parameter is the current hour time in the range 0..23. We are in trouble if the parrot is talking and the hour is before 7 or after 20. Return True if we are in trouble.

parrot_trouble(True, 6) → True
parrot_trouble(True, 7) → False
parrot_trouble(False, 6) → False

11. Write a program input Total Marks.
    If tm>360 then print 'First class'
    If Tm>=300 and tm<360 then print 'second class'
    If tm<300 then print 'third class'
12. We have two monkeys, a and b, and the parameters a_smile and b_smile indicate if each is smiling. We are in trouble if they are both smiling or if neither of them is smiling. Return True if we are in trouble
    monkey_trouble(True, True) → True
    monkey_trouble(False, False) → True
    monkey_trouble(True, False) → False
13. Given two int values, return their sum. Unless the two values are the same, then return double their sum.
    sum_double(1, 2) → 3
    sum_double(3, 2) → 5
    sum_double(2, 2) → 8
14. The parameter weekday is True if it is a weekday, and the parameter vacation is True if we are on vacation. We sleep in if it is not a weekday or we're on vacation. Return True if we sleep in.
    sleep_in(False, False) → True
    sleep_in(True, False) → False
    sleep_in(False, True) → True

15 What is the output?

```
if None:
    print("Hello")
```

2.The if...elif...else executes only one block of code among several blocks.

3. What is the output of the following code?

```
for i in [1, 0]:
  print(i+1)
```

4. In Python, for and while loop can have optional else statement?

5. What is the output of the following code?

```
i = sum = 0
while i <= 4:
   sum += i
   i = i+1
print(sum)
```

6. What is the output of the following code?

```
while 4 == 4:
      print('4')
```

7. Is it better to use for loop instead of while if you are iterating through a sequence (like: list)?

8. What is the output of the following code?

```
for char in 'PYTHON STRING':
  if char == ' ':
     break
  print(char, end='')
  if char == 'O':
     continue
```

Loop's Questions

15. Print Sathya Technologies for 10 times
16. Print 1 to 10 numbers
17. Print even numbers from 1 to 10
18. Print odd numbers from 1 to 10

19. Print sum of all the numbers 1 to 10
20. Print sum of all even numbers from 1 to 10
21. Print sum of all odd numbers from 1 to 10
22. Find the factorial of a given number
23. WAP to Check the number is prime or not
24. Print the number from 10 to 1
25. Input any 10 numbers find out no of even numbers and no of odd numbers
26. Input any 10 numbers find out no of positive numbers and no of negative numbers
27. Input any 10 numbers find the first biggest and second biggest number
28. Input any 10 find number of prime numbers.
29. Input any 4 digit number and reverse it
30. Input any 4 digit number and find out the sum of the digits
31. Input any 4 digit number and find out the sum of first and last digit
32. Input any 4 digit number and find out the sum of middle digits
33. Input any 4 digit number and find out the difference of all digits
34. Print the Multiplication table for any given Number.
35. What is the output of the following

```
string = "my name is Naveen"
for i in string:
      print (i, end=", ")
```

36. What is the output of the following

```
for i in range(10):
      if i == 5:
```

```
            break
        else:
            print(i)
    else:
        print("Here")
```

37. What is the output of the following

```
i = 0
while i < 3:
    print(i,end="")
    i += 1
else:
    print(0,end="")
```

38. Given a non-empty string like "Code" return a string like "CCoCodCode".
    string_splosion('Code') → 'CCoCodCode'
    string_splosion('abc') → 'aababc'
    string_splosion('ab') → 'aab'

39. Given a string, return a new string made of every other char starting with the first, so "Hello" yields "Hlo".
    string_bits('Hello') → 'Hlo'
    string_bits('Hi') → 'H'
    string_bits('Heeololeo') → 'Hello'

40. Given a string and a non-negative int n, we'll say that the front of the string is the first 3 chars, or whatever is there if the string is less than length 3. Return n copies of the front.
    front_times('Chocolate', 2) → 'ChoCho'
    front_times('Chocolate', 3) → 'ChoChoCho'
    front_times('Abc', 3) → 'AbcAbcAbc'

41. Given a string and a non-negative int n, return a larger string that is n copies of the original string.
    string_times('Hi', 2) → 'HiHi'
    string_times('Hi', 3) → 'HiHiHi'
    string_times('Hi', 1) → 'Hi'

42. Input any number check it is strong or not
    What is strong number: a number is called strong number if sum of the factorial of its digits is equal to number itself.
    Ex:145=1! + 4! + 5!=1+24+120

43. Input any number check it is palindrome
    What is palindrome: A word, phrase or sequence that reads the same backward as forward
    Ex:121,242,madam,

44. Input any number check armstrong or not
    What is Armstrong number: it is the sum of the cubes of its digits is equal to the number itself
    Ex:153=1^3+5^3+3^3=1+125+27=153

45. Input any number perfect number or not.
    What is perfect number it is a positive integer i.e equal to the sum of its proper positive divisors
    Ex : 6 Divisors of 6 are 1,2,3 1+2+3=6

Datatypes in Python

List

1) What is list
2) What is the purpose of list
3) Real time example on a list
4) What type of elements we can store in list

5) How to access list elements

6) What is forward and backward index

7) List allow duplicate objects

8) How to find length of a list

9) What operators we can use on list

10) Can we apply "+" operator on list and int

11) Can we apply "+" operator on list and list

12) If we apply "+" operator on list and list what will happen

13) What happens if we apply * operator on list

14)

L1=5	L1=[1,2,3]
L1*3	L1*3
Output ?	output ?

15) If we give out of index value in list, what happens

16) What is slice and define the Syntax

17) If we give invalid index in slicing expression what happens

18) What is the use of "in" operator in list

19) How to add an element at end of a list

20) How to insert an element in specific location in a list

21) How to arrange list element in ascending order

22) How to arrange list elements in descending order

23) How to remove an element in list

24) Difference between remove method , del statement and pop method

25) How to find lowest and highest values in a list

26) How to find sum of all elements in a list

27) How to compare two lists

28) How to add an element to empty list like

29) Can we copy one list to another list? If yes How.

30) What is two dimensional lists? When to use it

31) How to create two dimensional lists

32) Can we convert an integer object to list

33) What type of objects we can convert into list

34) Difference between append( ) and extend( )

35) How to clone or copy of a list

36) How to count occurrences of a list

37) How to split a list into evenly sized chunks

38) How to get an intersection of two list

39) How to remove duplicate elements in a list

40) How to create flat lists our of lists

41) Given 2 arrays of ints, a and b, return True if they have the same first element or they have the same last element. Both arrays will be length 1 or more.

Tuple

1) What is tuple
2) What is the purpose of tuple and when to use with a real time example
3) How to declare tuple
4) When to use tuple
5) Can we modify tuple data
6) Can we declare empty tuple
7) Can we declare a tuple with single element, if yes how, if no why
8) How to access tuples
9) What is packing and unpacking
10) Different types of methods in tuples
11) What is the use of count( ) method
12) What is the use of index( ) method
13) If I want to modify tuple, what to do
14) How to convert tuple to list
15) How to convert list to tuple
16) X=(10) x=(10,)
    Print(X*3) print(x*3)

Output : ? output : ?

SET

1) What is set
2) Can we store duplicate elements in sets
3) Can we modify set items
4) How to create sets
5) Can we create an empty sets by using { }
6) How to create empty sets
7) Can we access set elements by using index
8) Can we access set elements by using slice
9) How to access set elements
10) If we store duplicate elements in set what happens
11) How to add an element to set
12) Can we add new element in specific position
13) How to remove an element in set
14) How to work with remove( )
15) How to work with pop(
16) How to work with discard( )
17) What is the difference between discard( ) and remove( )
18) How to update set
19) What is the difference between add( ) and update( )

20) How to delete all the elements in a set

21) How to get common elements in two sets

22) What is meaning of " & " operator in sets

23) How to get all the elements in two sets

24) What is meaning of " | " operator in sets

25) How many ways to get difference in sets and explain

26) What is symmetric difference

27) Can we use for loop on sets to print set values

28) print(set("sathya")) output?

29) for x in set(" sathya "):
        print(x,end=" ")

30) Use of forzenset

31) How to convert any iterable object as set

Dictionary

1) What is Dictionary

2) When to use dictionary

3) How to store elements in dictionary

4) Can we declare duplicates elements as a keys in dictionary

5) Can we declare duplicate element as a value in dictionary

6) Can we declare empty set

7) Can we change dictionary

8) While performing modification , If key is not available then what happens

9) Can we change keys in dictionary

10) How to delete elements in dictionary

11) While deleting element in dictionary, if the key is not available what happens

12) How to avoid Exception

13) How to check length of a dictionary

14) Can we declare dictionary values as list

```
d1={  "Kohli"   : [14,85,145],
      "Dhoni"   : [120,67,50],
      "Rohit"   : [50,30,8],
      "Rahul"   : [110,90,56],
      "Rahane"  : [60,30,80]    }
```

15) Can we apply loop on dictionary? If yes how?

16) How to delete all the elements in dictionary

17) What is the use of get method

18) How to get all the keys in a dictionary

19) How to get deleted item in dictionary.

String

1) What is String

2) How to Access individual characters in a string

3) Given me a real time example for processing individual characters in strings

4) What's the index of first character in a string

5) If a string has 10 characters, then what's the index of last character

6) If we try to access a character which is out of range what happens

7) How can you find length of a string

8) qualification=" B.Tech" qualification[0]='M' What is the output

9) What is the meaning of immutable

10) Can we apply slicing on strings

11) If we give invalid index in slicing, what happens

12) How to check whether given string contains any symbols or not

13) How to check whether given string contains only alphabet or not

14) How to check whether given string contains only digits or not

15) How to check our string is in lower case or not

16) How to check our string is in upper case or not

17) How to check our string contains only whitespaces

18) How to convert our string in to lower case

19) How to delete spaces that are available in left side of a string

20) How to delete spaces that are available in right side of a string

21) How to delete spaces that are available in both left side and right side of a string

22) How to change existing string with a new string

23) How to find a string ends with specific string

24) How to check whether a given string starts with a specific string or not

Logical Questions on Datatypes

1. Given 2 arrays of ints, a and b, return True if they have the same first element or they have the same last element. Both arrays will be length 1 or more.

common_end([1, 2, 3], [7, 3]) → True
common_end([1, 2, 3], [7, 3, 2]) → False
common_end([1, 2, 3], [1, 3]) → True

2. Given an array of ints length 3, return the sum of all the elements.

3. Given an array of ints length 3, return a new array with the elements in reverse order, so {1, 2, 3} becomes {3, 2, 1}.

reverse3([1, 2, 3]) → [3, 2, 1]
reverse3([5, 11, 9]) → [9, 11, 5]
reverse3([7, 0, 0]) → [0, 0, 7]

4. Given an array of ints length 3, figure out which is larger, the first or last element in the array, and set all the other elements to be that value. Return the changed array.

max_end3([1, 2, 3]) → [3, 3, 3]
max_end3([11, 5, 9]) → [11, 11, 11]
max_end3([2, 11, 3]) → [3, 3, 3]

5. Given an array of ints, return the sum of the first 2 elements in the array. If the array length is less than 2, just sum up the elements that exist, returning 0 if the array is length 0.

sum2([1, 2, 3]) → 3
sum2([1, 1]) → 2
sum2([1, 1, 1, 1]) → 2

6. Given an array of ints length 3, return an array with the elements "rotated left" so {1, 2, 3} yields {2, 3, 1}.

rotate_left3([1, 2, 3]) → [2, 3, 1]
rotate_left3([5, 11, 9]) → [11, 9, 5]
rotate_left3([7, 0, 0]) → [0, 0, 7]

7. Given an int array length 2, return True if it contains a 2 or a 3.

has23([2, 5]) → True
has23([4, 3]) → True
has23([4, 5]) → False

8. Given an array of ints, return a new array length 2 containing the first and last elements from the original array. The original array will be length 1 or more.

make_ends([1, 2, 3]) → [1, 3]
make_ends([1, 2, 3, 4]) → [1, 4]
make_ends([7, 4, 6, 2]) → [7, 2]

9. Return the sum of the numbers in the array, returning 0 for an empty array. Except the number 13 is very unlucky, so it does not

count and numbers that come immediately after a 13 also do not count

sum13([1, 2, 2, 1]) → 6
sum13([1, 1]) → 2
sum13([1, 2, 2, 1, 13]) → 6

10. Given an array length 1 or more of ints, return the difference between the largest and smallest values in the array. Note: the built-in min(v1, v2) and max(v1, v2) functions return the smaller or larger of two values.

big_diff([10, 3, 5, 6]) → 7
big_diff([7, 2, 10, 9]) → 8
big_diff([2, 10, 7, 2]) → 8

11. Return the number of even ints in the given array. Note: the % "mod" operator computes the remainder, e.g. 5 % 2 is 1.

count_evens([2, 1, 2, 3, 4]) → 3
count_evens([2, 2, 0]) → 3
count_evens([1, 3, 5]) → 0

12. Given 2 int arrays, a and b, each length 3, return a new array length 2 containing their middle elements.

middle_way([1, 2, 3], [4, 5, 6]) → [2, 5]
middle_way([7, 7, 7], [3, 8, 0]) → [7, 8]
middle_way([5, 2, 9], [1, 4, 5]) → [2, 4]

13. Given an array of ints, return True if the array is length 1 or more, and the first element and the last element are equal.

same_first_last([1, 2, 3]) → False
same_first_last([1, 2, 3, 1]) → True
same_first_last([1, 2, 1]) → True

14. Return True if the string "cat" and "dog" appear the same number of times in the given string.

cat_dog('catdog') → True
cat_dog('catcat') → False
cat_dog('1cat1cadodog') → True

15. Return the number of times that the string "code" appears anywhere in the given string, except we'll accept any letter for the 'd', so "cope" and "cooe" count.

count_code('aaacodebbb') → 1
count_code('codexxcode') → 2
count_code('cozexxcope') → 2

16. Given two strings, return True if either of the strings appears at the very end of the other string, ignoring upper/lower case differences (in other words, the computation should not be "case sensitive"). Note: s.lower() returns the lowercase version of a string.

end_other('Hiabc', 'abc') → True
end_other('AbC', 'HiaBc') → True
end_other('abc', 'abXabc') → True

17. Return the number of times that the string "hi" appears anywhere in the given string

count_hi('abc hi ho') → 1
count_hi('ABChi hi') → 2
count_hi('hihi') → 2

18. Given a string, return a string where for every char in the original, there are two chars

double_char('The') → 'TThhee'
double_char('AAbb') → 'AAAAbbbb'
double_char('Hi-There') → 'HHii--TThheerree'

19. Given an array of ints, return True if 6 appears as either the first or last element in the array. The array will be length 1 or more.

first_last6([1, 2, 6]) → True
first_last6([6, 1, 2, 3]) → True
first_last6([13, 6, 1, 2, 3]) → False

1. What is the difference between list and tuples?
2. What is the difference between deep and shallow copy?
3. How can the ternary operators be used in python?
4. How is memory managed in Python?
5. Explain Inheritance in Python with an example.
6. Explain what Flask is and its benefits?
7. What is the usage of help() and dir() function in Python?
8. What is dictionary in Python?
9. What is monkey patching in Python?
10. What does this mean: *args, **kwargs? And why would we use it?

11. Write a one-liner that will count the number of capital letters in a file. Your code should work even if the file is too big to fit in memory.

12. What are negative indexes and why are they used?

13. How can you randomize the items of a list in place in Python?

14. What is the process of compilation and linking in python?

15. How can you generate random numbers in Python?

16. What is the difference between range & xrange?

17. What is pickling and unpickling?

Functions in Python

1) What is function → collection of statement that perform an operation
2) Advantage of function → easy to modify. Easy to debug
3) How functions make development faster Easy to modify, less code to write
4) How to define and call a function
5) What's the difference between define and calling a function by using def keyword & by using function name with () is calling a fun
6) Can we call a function inside of another function yes can
7) How many times we can call a function any time
8) What is local variable variable declared inside the fun or method
9) What is scope of local variable within the fun or method
10) Can we declare local variables with same name in different functions yes

11) How many ways we can define functions 4 ways
12) What is arguments in functions
13) What is parameter variables which are given in function or method
14) Can we pass multiple arguments yes
15) The modifications occur in parameters does it reflect to arguments no
16) What is named or keyword arguments passing values using parameter variable name
17) Can we mix non-keyword arguments with keyword arguments no must in order
18) What is a global variable → variable used by module, class, fun. or method
19) What is global constant → ??
20) What is the purpose of return statement in a function return value
21) What is Boolean function
22) Can we return multiple values no, multiple values are in tuple packing
23) What is lambda function
24) How to create Anonymous functions
25) How many arguments can we pass to lambda functions
26) Can we declare lambda functions with default arguments
27) What's the use of map in lambda
28) What's the use of filter function in lambda
29) What is use of reduce function
30) What is recursive function

Module

1) What is module
2) Can we call global variables or methods of one program to another program
3) How to access modules
4) How to call a function in side of another module
5) Can i change module names in our program
6) Can we import only required function in another module
7) Can we import two or more functions from module and how
8) I have a module, which contains variable ph, and two functions email( ) , ph( ). Now i want to import ph variable. How to do it.
9) How to know what are the functions and variables available in a module
10) Can we import more than one module at a time
11) Can we import required functions from more than one module

Packages:

1) What is package
2) To create a package, which file is compulsory
3) we need to write any logic for __init__.py file
4) How to access modules inside of a packages
5) Can we create package in side another package

6) we need to create __init__.py file inside sub packages
7) How to call sub-package related modules

OOPS

1. What is OOPS
2. How many types of principles in OOPS and what are they
3. What is OOPL
4. What is the difference between OOPS and OOP
5. Is Python supports OOPS
6. Is OOPS Principles fallows any order and what is that.
7. What is Abstraction
8. What is Encapsulation
9. How to achieve encapsulation in Python?
10. What is Inheritance
11. What is Polymorphism
12. What is class?
13. How to define a class?
14. What is Object?
15. What is the use of the Object?
16. How to create Object?
17. How to destroy or delete an Object from memory?
18. Where Object gets memory?

19. Where class gets memory?
20. What is instance variable?
21. What is static variable?
22. Why Instance variables?
23. Why static variables?
24. What is the difference between instance and static variables?
25. When static variables get memory?
26. When instance variables get memory?
27. Where static variables get memory?
28. Where instance variables get memory?
29. What are the ways of accessing static variables?
30. What are the ways of accessing instance variables?
31. What happens if I use all variables as instance in the class?
32. What happens if I use all variables as static in the class?
33. What is a method?
34. How to define a method in python?
35. How to use/call a method?
36. What is the use of method parameters?
37. What are the types of methods in python
38. When should I define an Instance method?

39. When should I define a static method?
40. What are the ways of accessing an instance method?
41. What are the ways of accessing a static method?
42. What are the variables can instance method access
43. What are the variables can static method access
44. Can we create a variable inside the method?
45. Can we print a variable without setting a value? If not why?
46. Who will provide the initial/default value to local variable?
47. Can we initialize the instance or static variables through method?
48. Can we assign the instance or static variables through method?
49. Who will provide the initial/default value to instance variable?
50. Can a method call another method?
51. Can I call a static method from instance method?
52. Can I call an instance method from instance method?
53. Can I call an instance method from static method?
54. Can I call a static method from static method?
55. What is constructor?
56. What are the rules in Defining a constructor?

57. Where/When Constructor is used?
58. Can I write a class without defining at least constructor?
59. Can I create an object if a class is having zero constructors?
60. What are the types of constructor?
61. What is the parameterized constructor?
62. When should we use parameterized constructor?
63. How to call a constructor?
64. Can we use a constructor to initialize static variables?
65. Can we call a constructor from a method(static/instance)?
66. Can we call a method from constructor?
67. Can constructors access static variables?
68. Can constructors access instance variables?
69. Can I declare constructor as static
70. Can I write a method with class name?
71. What happens if I define a method with class name?
72. What is Inheritance?
73. What is the use of Inheritance?
74. Can static members participate in inheritance?
75. Can instance members participate in inheritance?
76. Can we inherit a constructor?
77. What are the types of Inheritance?

78. What happens if we create an Object to sub class/any class?
79. How to initialize the super class instance variables?
80. How to call a super class constructor?
81. Who will call upper class constructor by default?
82. How to call one constructor from another constructor of same class?
83. What is the use of calling one constructor from another of same class?
84. What is Overloading?
85. What is method overloading?
86. What is constructor overloading?
87. What is the use of overloading?
88. Can we overload a constructor in different class?
89. Can we overload an instance method in different class?
90. Can we overload a constructor in same class?
91. Can we overload an instance method in same class?
92. Can we overload variables?
93. Can we overload static method?
94. Can we overload instance method?
95. What is overriding?
96. What is method overriding?

97. What is constructor overriding?
98. Is overriding done without inheritance?
99. What is the use of overriding?
100. Dose overriding depends on parameter?
101. Dose overriding depends on method name?
102. Can we overriding methods in same class?
103. Can we override methods in different class?
104. Can we override variables?
105. How to stop method overriding?

OOPS Case Studies

1. Write a class Define a class to represent a bank account. Include the following members:

Data Variables : Name of the Depositor, Account Number, Type of Account, Balance amount in the account.

Data Methods: To assign initial values, To deposit an amount, To withdraw an amount after checking the balance, To display name and balance.

2. Create a class called Student with the following details:

Roll No, Stud Name, Marks In Eng, Marks In Maths, Marks In Science. Display the total marks and Percentage of the student.

3. Change the above class definition so that you can calculate marks for five students.

4. For an Online Bookstore create a class to store book details and display the book details with fields are book number, book name, book title, book author, quantity of books, book price. calculate and display the bill amount .

5. Create a reference type called Person. Populate the Person class with the following attributes to store the following information: First name, Last name, Email address, Date of birth Add constructors that accept the following parameter lists: All four parameters First name , Last name , Email ,First name , Last name , Date of birth

write appropriate methods to accept and display the details.

6. Case Study:

Yatra.Com Travels is a committed tour and travel company. It has devised many innovative packages for its customers who want to take a holiday. There are three kinds of tours:

a)Discover India b)Holiday Hungama c)Pilgrimage Package

These tours start every week. A customer can avail of a package belonging to any category, starting on any given date. The customer can also specify the number of people accompanying the customer. The customer class will have the following member data: customer name ,number of people accompanying(int), package category(D/H/P), cost(float), tour start date

Write an executable program with a class called Customer with required attributes and methods.

7. Consider the following scenario:

A Furniture Manufacturer manufactures domestic furniture. Customers provide their specifications to the company for the

furniture they want. To cope up with the received customer's order, FFC decides to computerize the order-processing system. The System should accept the values of furniture items, such as a bookshelf and a chair. You need to develop the hierarchy of these items.

8. Create the classes required to store data regarding different types of Courses. All courses have name, duration and course fee. Some courses are part time where you have to store the timing for course. Some courses are onsite where you have to store the company name and the no. of candidates for the course. For onsite course we charge 10% more on the course fee. For part-time course, we offer 10% discount.

Provide constructors and the following methods.

Print() GetTotalFee()

9. Create a class to store details of student like rollno, name, course joined and feepaid so far. Assume courses are Core Pyhon and Advance Python with course fees being 3000 and 3500.

Provide a constructor to take rollno, name and course. Provide the following methods: Payment(amount) ,Print() ,DueAmount property, Total Fee property

Add a static member to store Service Tax, which is set to 12.3%. Also allow a property through which we can set and get service tax.

Modify Total Fee and Due Amount properties to consider service tax

10. Create a class called Customer with two methods

customertype() which displays the type of customer getPriviledge() which displays the privileges according to type of the customer.

Create a class called CorporateCustomer which inherits the Customer class and overides the methods given in the Customer class.

Create a class called PersonalCustomer which inherits the Customer class and overides the methods given in the Customer class.

create another class called MainClass to execute the program.

11. Write an executable program.

Create a class Shape with the following methods

getDetails() to get details from the user

calculateArea() to calculate the area with the given dimensions

displayDetails() to display the calculated area of the shape.

Create a class called Triangle which inherits Shape class and provides appropriate implementation of the methods given in the base class.

Create a class called Circle which inherits Shape class and provides appropriate implementation of the

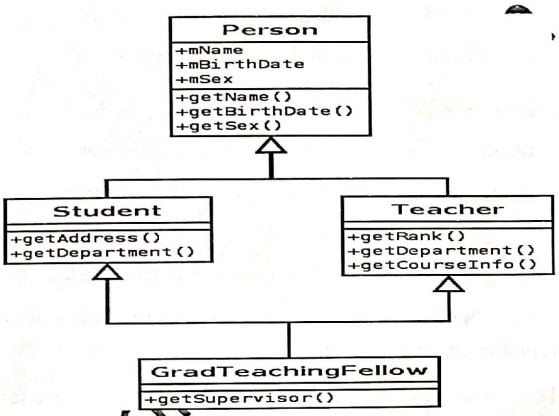
12. Martin wants to create a ticket booking application for a movie theater. The application should ask the user for his choice, whether he wants to book the tickets or not. The application should also ask the user for the total number of tickets to be booked. While booking the tickets if the total number of booked tickets exceeds the available tickets, the application should raise an exception.

Assume the total number of available tickets is 15.

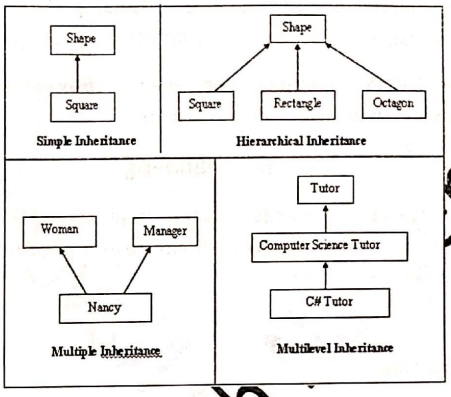
13. Create a base class, Telephone, and derive a class Electronic Phone from it. In Telephone, create a protected string member phonetype, and a public method Ring( ) that

outputs a text message like this: "Ringing the <phonetype>." In ElectronicPhone, the constructor should set the phonetype to "Digital." In the Run( ) method, call Ring( ) on the Electronic Phone to test the **inheritance**?

14. Implement the Following Multiple Inheritance Program using Interfaces



15. WAP to implement inheritance?



17. Create a class to implement shapes and draw the following shapes

a) Rectangle

b) Circle

c) Triangle

And Find their Areas Using Method Area.

18. Create a class to implement calculate simple and compound interest using method find interest method.

19. Create a class to Calculate Employee Total Salary Accept Basic Salary, DA, HRA from keyboard If Employee Salary Greater Then 20,000 Then increment 20% of salary and Display Total Salary

Overloading & overriding:

20.Wap on Overloading to Implement Addition operations with different Parameters ex : add(int,int), add(float,int), add(float,int,str);

21. Wap on Overriding Methods To implement Display method In parent class and child class with same signature.

Exception Handling

1. How many except statements can a try-except block have?
2. When will the else part of try-except-else be executed?
3. Is the following code valid?

```
try:
    # Do something
except:
    # Do something
finally:
    # Do something
```

4. Is the following code valid?

```
try:
    # Do something
except:
    # Do something
else:
    # Do something
```


5. Can one block of except statements handle multiple exception?
6. When is the finally block executed?
7. What is the output of the following code?

```
def foo():
    try:
        return 1
    finally:
        return 2
k = foo()
print(k)
```

8. What is the output of the following code?

```
def foo():
    try:
        print(1)
    finally:
        print(2)
foo()
```

9. What is the output of the following?

```
try:
    if '1' != 1:
        raise "someError"
    else:
```

```
    print("someError has not occured")
except "someError":
    print ("someError has occured")
```

10. What happens when '1' == 1 is executed?
11. Which of the following is not a standard exception in Python?

a) NameError
b) IOError
c) AssignmentError
d) ValueError

12. Syntax errors are also known as parsing errors.

13. An exception is object

14. Which of the following blocks will be executed whether an exception is thrown or not?

a) except
b) else
c) finally
d) assert

Python File Input Output

1. Write a Python program to read an entire text file.

2. Write a Python program to read first n lines of a file.

3. Write a Python program to append text to a file and display the text.

4. Write a Python program to read last n lines of a file.

5. Write a Python program to read a file line by line and store it into a list.

6. Write a Python program to read a file line by line store it into a variable.

7. Write a Python program to read a file line by line store it into an array

8. Write a python program to find the longest words.

9. Write a Python program to count the number of lines in a text file.

10. Write a Python program to count the frequency of words in a file.

11. Write a Python program to get the file size of a plain file.

12. Write a Python program to write a list to a file.

13. Write a Python program to copy the contents of a file to another file.

14. Write a Python program to combine each line from first file with the corresponding line in second file.

15. Write a Python program to read a random line from a file.

16. Write a Python program to assess if a file is closed or not.

17. Write a Python program to remove newline characters from a file.

18. What Will Be The Output Of The Following Code Snippet?

```
fo = open("myfile.txt", "w+")
print ("Name of the file: ", fo.name)
# Assuming that the file contains these lines
# SathyaTech
```

```
# This is Naveen Kumar
seq=" SathyaTech\ This is Naveen Kumar"
fo.writelines(seq )
fo.seek(0,0)
for line in fo:
   print (line)
fo.close()
```

19. What Will Be The Output Of The Following Code Snippet?

```
import sys
print ('Enter your name: ',)
name = ''
while True:
  c = sys.stdin.read(1)
  if c == '\n':
    break
  name = name + c
print ('Entered name is:', name)
```

Assuming that input given by user is : Naveen kumar

20. What Will Be The Output Of The Following Code Snippet?

```
with open("hello.txt", "w") as f:
   f.write("Hello World how are you today")
```



```
with open('hello.txt', 'r') as f:
  data = f.readlines()
  for line in data:
     words = line.split()
     print (words)
  f.close()
```

21. What Will Be The Output Of The Following Code Snippet?

```
f = None
for i in range (5):
   with open("myfile.txt", "w") as f:
      if i > 2:
         break
print (f.closed)
```

22. What Will Be The Output Of The Following Code Snippet?

```
f = open("data.txt", "r")
txt = "This is 1st line\n"
f.writelines( txt )
f.seek(0,0)
line = f.readlines()
print ("Read Line: %s" % (line))
```

```
f.close()
```

23. What Will Be The Output Of The Following Code Snippet?

```
try:
  f = open("testfile", "r")
  f.write("This is the test file for exception handling!!")
except IOError:
  print ("Error: could not find a file or read data")
else:
  print ("content is written in the file successfully")
```

24. What Will Be The Output Of The Following Code Snippet?

```
colors = ['red\n', 'yellow\n', 'blue\n']
f = open('colors.txt', 'w')
f.writelines(colors)
f.close()
f.seek(0,0)
for line in f:
  print (line)
```


Regular Expressions

Regular expressions are also called as REs, or regexes, or regex.

Regular expressions are available in re module.

Using Regular expressions we can specify the rules for the set of possible strings that you want to match; this set might contain English sentences, or e-maii addresses, or TeX commands, or anything you like.

Various methods of Regular Expressions?

1. **re.match()**
2. **re.search()**
3. **re.findall()**
4. **re.split()**
5. **re.sub()**
6. **re.compile()**

re.match(*pattern,string*):

This method finds match if it occurs at start of the string.

Example:

```
>>> import re
>>> st = "This is Naveen"
>>> result = re.match(r"Naveen",st)
>>> print(result)
```

Output : None

Example

```
>>> import re
>>> st = "Naveen is here"
>>> result = re.match(r"Naveen",st)
>>> print(result)
```

output : <_sre.SRE_Match object; span=(0, 6), match='Naveen'>

Note: To print the matching string we'll use method group() (It helps to return the matching string).

Example

```
>>> import re
>>> st = "Naveen is here"
>>> result = re.match(r"Naveen",st)
>>> print(result.group(0))
```

Output : Naveen

There are methods like start() and end() to know the start and end position of matching pattern in the string.

Example

```
>>> import re
>>> st = "Naveen is from Sathya Naveen is teaching Python"
>>> result = re.match(r"Naveen",st)
>>> print(result.start())
```

Output: 0


```
>>> print(result.end())
```

Output: 6

re.search(*pattern*, *string*):

It is similar to match() but it doesn't restrict us to find matches at the beginning of the string only.

Example:

```
>>> import re
>>> st = "This is Naveen"
>>> result = re.search(r"Naveen",st)
>>> print(result)
```

Output : <_sre.SRE_Match object; span=(8, 14), match='Naveen'>

```
>>> print(result.group(0))
```

Output: Naveen

re.findall (*pattern*, *string*):

It helps to get a list of all matching patterns. It has no constraints of searching from start or end.

Example:

```
>>> import re
>>> st = "Naveen is from Sathya Naveen is teaching Python"
>>> result = re.findall(r"Naveen",st)
>>> print(result)
```

Output: ['Naveen', 'Naveen']

re.split(*pattern*, *string*, [*maxsplit=0*]):

This method helps to split *string* by the occurrences of given *pattern*.

Example

```
>>> import re
>>> st = "kumar"
>>> result = re.split(r"m",st)
>>> print(result)
```

Output : ['ku', 'ar']

Example

```
>>> import re
>>> st = "Programming"
>>> result = re.split(r"m",st)
>>> print(result)
```

Output: ['Progra', 'ing']

Method split() has another argument "maxsplit". It has default value of zero. In this case it does the maximum splits that can be done, but if we give value to maxsplit, it will split the string.

Example:

```
>>> import re
>>> st = "Programming"
>>> result = re.split(r"m",st,maxsplit=1)
```



```
>>> print(result)
```

Output: ['Progra', 'ming']

re.sub(*pattern*, *repl*, *string*):

It helps to search a pattern and replace with a new sub string. If the pattern is not found, *string* is returned unchanged.

Example

```
>>> import re
>>> st = "Current rocking programming is Java"
>>> result = re.sub(r"Java","Python",st)
>>> print(result)
```

Output: Current rocking programming is Python

What are the most commonly used operators?

Get the Complete ref @ https://docs.python.org/2/library/re.html

Operators	Description
.	Matches with any single character except newline '\n'.
?	match 0 or 1 occurrence of the pattern to its left
+	1 or more occurrences of the pattern to its left
*	0 or more occurrences of the pattern to its left
\w	Matches with a alphanumeric character whereas \W (upper case W) Matches non alphanumeric character.
\d	Matches with digits [0-9] and /D (upper case D) matches with non-digits.
\s	Matches with a single white space character (space, newline, return, tab, form) and \S (upper case S) matches any non-white space character.
\b	boundary between word and non-word and /B is opposite of

	/b
[..]	Matches any single character in a square bracket and [^..] matches any single character not in square bracket
\	It is used for special meaning characters like \. to match a period or \+ for plus sign.
^ and $	^ and $ match the start or end of the string respectively
{n,m}	Matches at least n and at most m occurrences of preceding expression if we write it as {,m} then it will return at least any minimum occurrence to max m preceding expression.
a| b	Matches either a or b
( )	Groups regular expressions and returns matched text
\t, \n, \r	Matches tab, newline, return

Problem 1: Return the first word of a given string

Solution-1 Extract each character (using "\w")

```
>>> import re
>>> s1 = "I am learning python with Naveen"
>>> result = re.findall(r".",s1)
>>> print(result)
```

Output: ['I', ' ', 'a', 'm', ' ', 'l', 'e', 'a', 'r', 'n', 'i', 'n', 'g', ' ', 'p', 'y', 't', 'h', 'o', 'n', ' ', 'w', 'i', 't', 'h', ' ', 'N', 'a', 'v', 'e', 'e', 'n']

Above space is also extracted, now to avoid it use "\w" instead of ".".

```
>>> import re
>>> s1 = "I am learning python with Naveen"
>>> result = re.findall(r"\w",s1)
>>> print(result)
```


Output: ['I', 'a', 'm', 'l', 'e', 'a', 'r', 'n', 'i', 'n', 'g', 'p', 'y', 't', 'h', 'o', 'n', 'w', 'i', 't', 'h', 'N', 'a', 'v', 'e', 'e', 'n']

Solution-2 Extract each word (using "*" or "+")

```
>>> import re
>>> s1 = "I am learning python with Naveen"
>>> result = re.findall(r"\w*",s1)
>>> print(result)
```

output: ['I', '', 'am', '', 'learning', '', 'python', '', 'with', '', 'Naveen', '']

Again, it is returning space as a word because "*" returns zero or more matches of pattern to its left. Now to remove spaces we will go with "+".

```
>>> import re
>>> s1 = "I am learning python with Naveen"
>>> result = re.findall(r"\w+",s1)
>>> print(result)
```

output: ['I', 'am', 'learning', 'python', 'with', 'Naveen']

Solution-3 Extract each word (using "^")

```
>>> import re
>>> s1 = "I am learning python with Naveen"
>>> result = re.findall(r"^\w+",s1)
>>> print(result)
```

Output: ['I']

If we will use "$" instead of "^", it will return the word from the end of the string. Let's look at it.

```
>>> import re
>>> s1 = "I am learning python with Naveen"
>>> result = re.findall(r"\w+$",s1)
>>> print(result)      Output : ['Naveen']
```

Problem 2: Return the first two character of each word

Solution-1 Extract consecutive two characters of each word, excluding spaces (using "\w")

```
>>> import re
>>> s1 = "I am learning python with Naveen"
>>> result = re.findall(r"\w\w",s1)
>>> print(result)
```

Output: ['am', 'le', 'ar', 'ni', 'ng', 'py', 'th', 'on', 'wi', 'th', 'Na', 've', 'en']

Solution-2 Extract consecutive two characters those available at start of word boundary (using "\b")

```
>>> import re
>>> s1 = "I am learning python with Naveen"
>>> result = re.findall(r"\b\w.",s1)
>>> print(result)
```

Output: ['I ', 'am', 'le', 'py', 'wi', 'Na']

Problem 3: Return the domain type of given email-ids

Solution-1 Extract all characters after "@"

```
>>> import re
>>> s1 = "naveen@gmail.com mailmenaveenkumar@yahoo.com pythonnaveen@co.in"
>>> result = re.findall(r"@\w+",s1)
>>> print(result)
```

Output: ['@gmail', '@yahoo', '@co']

Above, you can see that ".com", ".in" part is not extracted. To add it, we will go with below code.

```
>>> import re
>>> s1 = "naveen@gmail.com mailmenaveenkumar@yahoo.com pythonnaveen@co.in"
>>> result = re.findall(r"@\w+.\w+",s1)
>>> print(result)
```

['@gmail.com', '@yahoo.com', '@co.in']

Solution – 2 Extract only domain name using "( )"

```
>>> import re
>>> s1 = "naveen@gmail.com mailmenaveenkumar@yahoo.com pythonnaveen@co.in"
>>> result = re.findall(r"@\w+.(\w+)",s1)
>>> print(result)
```

Output: ['com', 'com', 'in']

Problem 4: Return date from given string

Here we will use "\d" to extract digit.

```
>>> import re
>>> s1 = "Hi 007 this is 001 from python"
>>> result = re.findall(r"\d",s1)
>>> print(result)
```

Output: ['0', '0', '7', '0', '0', '1']

Problem 5: Return all words of a string those starts with vowel

Solution-1 Return each words

```
>>> import re
>>> s1 = "I am learning python with Naveen"
>>> result = re.findall(r"\w+",s1)
>>> print(result)
```

Output: ['I', 'am', 'learning', 'python', 'with', 'Naveen']

Solution-2 Return words starts with alphabets (using [])

```
>>> import re
>>> s1 = "I am learning python with Naveen"
>>>result = re.findall(r"[aeiouAEIOU]\w+",s1)
>>> print(result)
```

['am', 'earning', 'on', 'ith', 'aveen']

Example to validate Name

```
import re
user_name = input("Name Please  :")
result = re.match("^[A-Za-z]*$", user_name)
if result == None:
  print("Invalid Name")
else:
  print("Welcome Mr/Miss : ",user_name)
```


Match a string to a numberic sequence of exactly five

```
import re
input = input("Enter an input string:")
m = re.match('\d{5}\Z',input)
if m:
    print("True")
else:
    print("False")
```

Email validation regex

```
import re
input = input("Enter an input string:")
m = re.match('[^@]+@[^@]+\.[^@]+',input)
if m:
  print("True")
else:
  print("False")
```

To search if an e-mail address is in a string:

```
import re
input = "Contact me by mailmenaveenkumar@gmail.com or at the office."
m = re.search('[^@]+@[^@]+\.[^@]+',input)
if m:    print("String found.")
else:    print("Nothing found.")
```

Mathematical Functions

The math module is a standard module in Python and is available by default. To use mathematical functions under this module, you have to import the module using import math.

Example

Square root calculation

```
import math
math.sqrt(4)
```

Note: This module does not support complex datatypes

Function	Description
ceil(x)	Returns the smallest integer greater than or equal to x.
copysign(x, y)	Returns x with the sign of y
fabs(x)	Returns the absolute value of x
factorial(x)	Returns the factorial of x

floor(x)	Returns the largest integer less than or equal to x
fmod(x, y)	Returns the remainder when x is divided by y
frexp(x)	Returns the mantissa and exponent of x as the pair (m, e)
fsum(iterable)	Returns an accurate floating point sum of values in the iterable
isfinite(x)	Returns True if x is neither an infinity nor a NaN (Not a Number)
isinf(x)	Returns True if x is a positive or negative infinity
isnan(x)	Returns True if x is a NaN
ldexp(x, i)	Returns x * (2**i)
modf(x)	Returns the fractional and integer parts of x
trunc(x)	Returns the truncated integer value of x
exp(x)	Returns e**x
expm1(x)	Returns e**x - 1
log(x[, base])	Returns the logarithm of x to the base (defaults to e)
log1p(x)	Returns the natural logarithm of 1+x
log2(x)	Returns the base-2 logarithm of x

log10(x)	Returns the base-10 logarithm of x
pow(x, y)	Returns x raised to the power y
sqrt(x)	Returns the square root of x
acos(x)	Returns the arc cosine of x
asin(x)	Returns the arc sine of x
atan(x)	Returns the arc tangent of x
atan2(y, x)	Returns atan(y / x)
cos(x)	Returns the cosine of x
hypot(x, y)	Returns the Euclidean norm, sqrt(x*x + y*y)
sin(x)	Returns the sine of x
tan(x)	Returns the tangent of x
degrees(x)	Converts angle x from radians to degrees
radians(x)	Converts angle x from degrees to radians
acosh(x)	Returns the inverse hyperbolic cosine of x
asinh(x)	Returns the inverse hyperbolic sine of x
atanh(x)	Returns the inverse hyperbolic tangent of x

cosh(x)	Returns the hyperbolic cosine of x
sinh(x)	Returns the hyperbolic cosine of x
tanh(x)	Returns the hyperbolic tangent of x
erf(x)	Returns the error function at x
erfc(x)	Returns the complementary error function at x
gamma(x)	Returns the Gamma function at x
lgamma(x)	Returns the natural logarithm of the absolute value of the Gamma function at x
pi	Mathematical constant, the ratio of circumference of a circle to it's diameter (3.14159...)
e	mathematical constant e (2.71828...)

Loops

1. Write a Python program to find those numbers which are divisible by 7 and multiple of 5, between 1500 and 2700 (both included

2. Write a Python program to guess a number between 1 to 9.

Note : User is prompted to enter a guess. If the user guesses wrong then the prompt appears again until the guess is correct, on

successful guess, user will get a "Well guessed!" message, and the program will exit.

4. Write a Python program to construct the following pattern, using a nested for loop.

```
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
```

5. Write a Python program that accepts a word from the user and reverse it.

6. Write a Python program to count the number of even and odd numbers from a series of numbers.
Sample numbers : numbers = (1, 2, 3, 4, 5, 6, 7, 8, 9)
Expected Output :
Number of even numbers : 5
Number of odd numbers : 4

7. Write a Python program that prints each item and its corresponding type from the following list.
Sample List : datalist = [1452, 11.23, 1+2j, True, 'Sathya', (0, -1), [5, 12], {"class":'V', "section":'A'}]

8. Write a Python program that prints all the numbers from 0 to 6 except 3 and 6.

Note : Use 'continue' statement.
Expected Output : 0 1 2 4 5

9. Write a Python program to get the Fibonacci series between 0 to 50. Note : The Fibonacci Sequence is the series of numbers :
0, 1, 1, 2, 3, 5, 8, 13, 21, ...
Every next number is found by adding up the two numbers before it.
Expected Output : 1 1 2 3 5 8 13 21 34

10. Write a Python program which iterates the integers from 1 to 50. For multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".
Sample Output :
fizzbuzz
1
2
fizz
4
buzz

11. Write a Python program which takes two digits m (row) and n (column) as input and generates a two-dimensional array. The element value in the i-th row and j-th column of the array should be i*j.

Note :
i = 0,1.., m-1
j = 0,1, n-1.

Test Data : Rows = 3, Columns = 4
Expected Result : [[0, 0, 0, 0], [0, 1, 2, 3], [0, 2, 4, 6]]

12. Write a Python program that accepts a sequence of lines (blank line to terminate) as input and prints the lines as output (all characters in lower case).

13. Write a Python program which accepts a sequence of comma separated 4 digit binary numbers as its input and print the numbers that are divisible by 5 in a comma separated sequence.
Sample Data : 0100,0011,1010,1001,1100,1001
Expected Output : 1010

14. Write a Python program that accepts a string and calculate the number of digits and letters.

Sample Data : Python 3.2
Expected Output :
Letters 6
Digits 2

15. Write a Python program to check the validity of password input by users.
Validation :

- At least 1 letter between [a-z] and 1 letter between [A-Z].
- At least 1 number between [0-9].
- At least 1 character from [$#@].
- Minimum length 6 characters.
- Maximum length 16 characters.

16. Write a Python program to find numbers between 100 and 400 (both included) where each digit of a number is an even number. The numbers obtained should be printed in a comma-separated sequence.

17. Write a Python program to print alphabet pattern 'A'.
Expected Output:

```
***
* *
* *
*****
* *
* *
* *
```

18. Write a Python program to print alphabet pattern 'D'.
Expected Output:

```
****
* *
* *
* *
* *
****
```

19. Write a Python program to print alphabet pattern 'E'.
Expected Output:

```
*****
*
```

```
 *
 ****
 *
 *
 *****
```

20. Write a Python program to print alphabet pattern 'G'.

Expected Output:

```
 ***
* *
*
* ***
* *
* *
 ***
```

21. Write a Python program to print alphabet pattern 'L'.

Expected Output:

```
*
*
*
*
*
*
```



```
*****
```

22. Write a Python program to print alphabet pattern 'M'.

Expected Output:

```
*   *
*   *
** **
* * *
*   *
*   *
*   *
```

23. Write a Python program to print alphabet pattern 'O'.

Expected Output:

```
 ***
* *
* *
* *
*  *
* *
 ***
```

24. Write a Python program to print alphabet pattern 'P'.

Expected Output:

```
****
```

```
* *
* *
****
*
*
*
```

25. Write a Python program to print alphabet pattern 'R'.

Expected Output:

```
****
* *
* *
****
* *
* *
* *
```


26. Write a Python program to print the following patterns

Expected Output:

```
 ****
 *
 *
```



```
***
 *
 *
****
```

27. Write a Python program to print alphabet pattern 'T'.

Expected Output:

```
*****
  *
  *
  *
  *
  *
```


28. Write a Python program to print alphabet pattern 'U'.

Expected Output:

```
* *
* *
* *
* *
* *
* *
 ***
```

29. Write a Python program to print alphabet pattern 'X'.

Expected Output:

```
* *
* *
 **
 *
 **
* *
* *
```

30. Write a Python program to print alphabet pattern 'Z'.

Expected Output:

```
*******
   *
  *
 *
*
*******
```

31. Write a Python program to calculate a dog's age in dog's years.
Note: For the first two years, a dog year is equal to 10.5 human years. After that, each dog year equals 4 human years.
Expected Output:

Input a dog's age in human years: 15

The dog's age in dog's years is 73

32. Write a Python program to check whether an alphabet is a vowel or consonant.
Expected Output:

Input a letter of the alphabet: k

k is a consonant.

33. Write a Python program to convert month name to a number of days.

Expected Output:

List of months: January, February, March, April, May, June, July, August

, September, October, November, December

Input the name of Month: February

No. of days: 28/29 days

34. Write a Python program to sum of two given integers. However, if the sum is between 15 to 20 it will return 20.

35. Write a Python program to check a string represent an integer or not.

Expected Output:

Input a string: Python

The string is not an integer.

36. Write a Python program to check a triangle is equilateral, isosceles or scalene.

Input a dog's age in human years: 15

The dog's age in dog's years is 73

32. Write a Python program to check whether an alphabet is a vowel or consonant.

Expected Output:

Input a letter of the alphabet: k

k is a consonant.

33. Write a Python program to convert month name to a number of days.

Expected Output:

List of months: January, February, March, April, May, June, July, August

, September, October, November, December

Input the name of Month: February

No. of days: 28/29 days

34. Write a Python program to sum of two given integers. However, if the sum is between 15 to 20 it will return 20.

35. Write a Python program to check a string represent an integer or not.

Expected Output:

Input a string: Python

The string is not an integer.

36. Write a Python program to check a triangle is equilateral, isosceles or scalene.

Note :

An equilateral triangle is a triangle in which all three sides are equal.

A scalene triangle is a triangle that has three unequal sides.

An isosceles triangle is a triangle with (at least) two equal sides.

Expected Output:

Input lengths of the triangle sides:

x: 6

y: 8

z: 12

Scalene triangle

37. Write a Python program that reads two integers representing a month and day and prints the season for that month and day.

Expected Output:

Input the month (e.g. January, February etc.): july

Input the day: 31

Season is autumn.

38. Write a Python program to display astrological sign for given date of birth.

Expected Output:

Input birthday: 15

Input month of birth (e.g. march, july etc): may

Your Astrological sign is : Taurus

Note :
An equilateral triangle is a triangle in which all three sides are equal.
A scalene triangle is a triangle that has three unequal sides.
An isosceles triangle is a triangle with (at least) two equal sides.
Expected Output:

Input lengths of the triangle sides:

x: 6

y: 8

z: 12

Scalene triangle

37. Write a Python program that reads two integers representing a month and day and prints the season for that month and day.
Expected Output:

Input the month (e.g. January, February etc.): july

Input the day: 31

Season is autumn

38. Write a Python program to display astrological sign for given date of birth.
Expected Output:

Input birthday: 15

Input month of birth (e.g. march, july etc): may

Your Astrological sign is : Taurus

39. Write a Python program to display the sign of the Chinese Zodiac for given year in which you were born.
Expected Output:

Input your birth year: 1973

Your Zodiac sign : Ox

40. Write a Python program to find the median of three values.
Expected Output:

Input first number: 15

Input second number: 26

Input third number: 29

The median is 26.

41. Write a Python program to get next day of a given date.
Expected Output:

Input a year: 2016

Input a month [1-12]: 08

Input a day [1-31]: 23

The next date is [yyyy-mm-dd] 2016-8-24

42. Write a Python program to calculate the sum and average of n integer numbers (input from the user). Input 0 to finish

43. Write a Python program to create the multiplication table (from 1 to 10) of a number.
Expected Output:

Input a number: 6

6 x 1 = 6

6 x 2 = 12

6 x 3 = 18

6 x 4 = 24

6 x 5 = 30

6 x 6 = 36

6 x 7 = 42

6 x 8 = 48

6 x 9 = 54

6 x 10 = 60

44. Write a Python program to construct the following pattern, using a nested loop number

Expected Output:

1

22

333

4444

55555

666666

7777777

88888888

999999999