**Introduction to Python Syllabus**

**Day 1**

1. **Introduction to Python**. An introduction to the Python programming language. Covers details of how to start and stop the interpreter and write programs. Introduces Python's basic datatypes, files, functions, and error handling.
2. **Working with Data**. A detailed tour of how to represent and work with data in Python. Covers tuples, lists, dictionaries, and sets. Students will also learn how to effectively use Python's very powerful list processing primitives such as list comprehensions. Finally, this section covers critical aspects of Python's underlying object model including variables, reference counting, copying, and type checking.
3. **Program Organization and Functions**. More information about how to organize larger programs into functions. A major focus of this section is on how to design functions that are reliable and can be easily reused in other settings. Also covers technical details of functions including scoping rules, documentation strings, and anonymous functions.

**Day 2**

1. **Modules and Libraries**. How to organize programs into modules and details on using modules as a tool for creating extensible programs. Concludes with a tour of some of the most commonly used library modules including user interfaces and database integration. Also includes information on how to install third-party library modules.
2. **Classes and Objects**. An introduction to object-oriented programming in Python. Describes how to create new objects, overload operators, and utilize Python special methods. Also covers basic principles of object oriented programming including inheritance and composition.
3. **Inside the Python Object System**. A detailed look at how objects are implemented in Python. Major topics include object representation, attribute binding, inheritance, memory management, and special properties of classes including properties, slots, and private attributes.
4. **Testing, Debugging, and Software Development Practice**. This section discusses many isses that are considered important to Python software development. This includes effective use of documentation strings, program testing using both the doctest and unittest modules, and effective use of assertions. The Python debugger and profiler are also described.

**Day 3**

1. **Iterators and Generators**. Covers the iteration protocol, iterable objects, generators and generator expressions. A major focus of this section concerns the use of generators to set up data processing pipelines--a particularly effective technique for addressing a wide variety of common systems programming problems (e.g., processing large data files, handling infinite data streams, etc.).
2. **Working with Text**. A more in-depth look at how to efficiently parse and generate text data in Python. Topics include string splitting, string stripping, string joining, template strings, and Unicode. This section also covers the re module, Python's module for regular expression pattern matching and substitution.
3. **Binary Data Handling** Details on binary I/O and library modules for handling binary-encoded data (struct, array, etc.).
4. **Working with Processes**. This section covers details of the Python interpreter process including command line options and environment variables. The problem of interacting with other processes and applications from Python is then discussed with detailed coveraged of the subprocess module.
5. **Python Integration Primer**. A survey of how Python is able to interact with programs written in other programming languages. Topics include support for common data interchange formats (e.g., XML), network programming, accessing C code, COM extensions, Jython, and IronPython.

**LIST**

List is a sequence data type that holds group of elements in the form of array. But the main difference between array and list is array holds homogenous elements but list holds heterogeneous elements.

List has features in python like

1. It can expand and shrink up to the requirement as it is dynamic in nature. It is mutable.
2. It can take heterogeneous and homogenous elements as well.
3. We can also take an empty list.
4. List stores elements in insertion order.
5. List can hold duplicate elements.
6. It is represented through [] brackets.
7. All the elements are separated by coma(,).

**Processing of List and It’s elements**

1. Take an empty list
2. Create a list with different elements.
3. Create list using range() function.
4. Access list elements using loops.
5. Display list elements in different order manually.
6. Update list elements manually.
7. Concatenate lists.
8. Repetition of list or multiply list with an integer
9. Membership in list.
10. Aliasing and cloning of List.
11. Find biggest and smallest elements in list using methods and manually.
12. Which methods can be applied to list.
13. Use of list() method
14. Use of Index() method
15. Use of append() method
16. Use of insert() method
17. Use of remove() method
18. Use of copy() method
19. Use of extend() method
20. Use of pop() method
21. Use of cmp() method
22. Use of sort() method and different types of sorting manually
23. Use of reverse() method
24. Use of clear() method
25. Use of count() method
26. Use of round() method
27. Use of enumerate() method
28. Use of reduce() method
29. Use of sum() method
30. Use of ord() method
31. Use of all() method
32. Use of any() method
33. Use of accumulate() method
34. Use of filter() method
35. Use of map() method
36. Use of lambda() method
37. Find number of occurrences of element in list.
38. Find common elements of lists using sets and manually.
39. Storing different types of data manually into list
40. List comprehensions.
41. Nested list.
42. Use nested list as matrices.

**Other Programs**

1. Python Slicing
2. Find most frequent element in a list.
3. List comprehension and ord() in Python
4. Python List Comprehensions vs Generator Expressions
5. List Methods in Python – Set 1 Set 2
6. Lambda expression and filter function
7. Numbers in a list within a given range
8. Three way partitioning of an array around a given range
9. Sort even-placed elements in increasing and odd-placed in decreasing order
10. Segregate 0’s and 1’s in an array list
11. Python map function to find row with maximum number of 1’s
12. Print list after removing element at given index
13. Check if all the values in a list that are greater than a given value
14. Comprehension to find pair with given sum from two arrays
15. Get unique values from a list
16. Prefix sum array in Python using accumulate function
17. Check whether two lists are circularly identical
18. Check if two lists have at-least one element common
19. Difference between two lists
20. Maximum and minimum element’s position in a list
21. Print first m multiples of n without using any loop in Python
22. Program to check if two given matrices are identical
23. Sort the values of first list using second list
24. Print all the common elements of two lists
25. Print all sublists of a list
26. Iterate over multiple lists simultaneously
27. Count occurrences of an element in a list
28. Count set bits using Python List comprehension
29. Creating a 3D List
30. Creating a sorted merged list of two unsorted lists in Python
31. Largest, Smallest, Second Largest, Second Smallest in a List
32. Find average of a list in python
33. Sort a List according to the Length of the Elements
34. Multiply all numbers in the list
35. Generate random numbers within a given range and store in a list
36. Union of two or more Lists
37. Intersection of two lists
38. Remove and print every third from list until it becomes empty
39. Print anagrams together in Python using List and Dictionary
40. Move all zeroes to end of array using List Comprehension in Python
41. Sort a list according to the second element in sublist
42. Convert an array to an ordinary list with the same items
43. Python program to find Cumulative sum of a list
44. Find missing and additional values in two lists
45. Split the Even and Odd elements into two different lists
46. Python program to right rotate a list by n
47. All palindrome numbers in a list
48. Program to print duplicates from a list of integers
49. Find all close matches of input string from a list
50. Remove empty tuples from a list
51. Maximum sum of elements of list in a list of lists
52. Permutation of a given string using inbuilt function
53. Python program to find N largest elements from a list
54. Break a list into chunks of size N in Python
55. Program to convert String to a List
56. Using List as Stack and Queues in Python
57. Ways to sort list of dictionaries by values in Python – Using itemgetter

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 | 110 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 10 | 20 | 30 | 40 | 50 |

|  |
| --- |
| 120 |