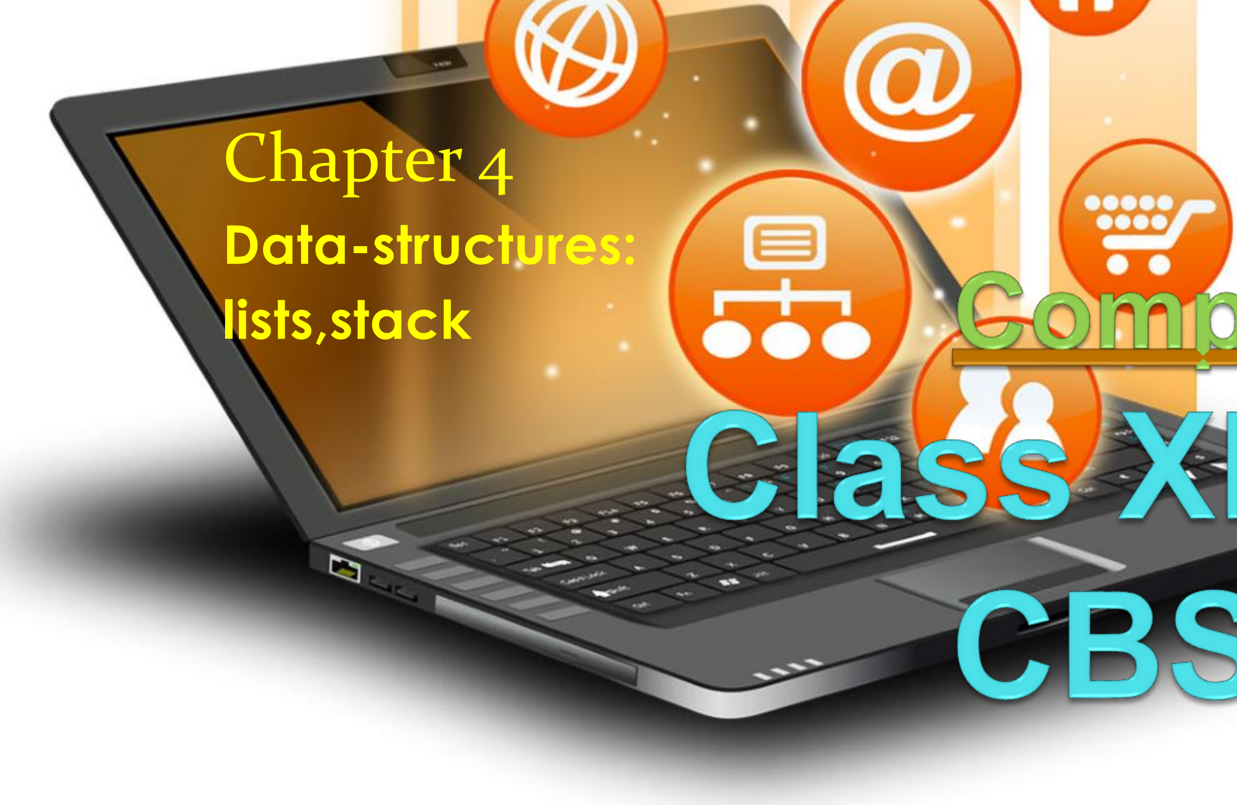


New syllabus 2022-23

Chapter 4

Data-structures: lists,stack

Computer Science

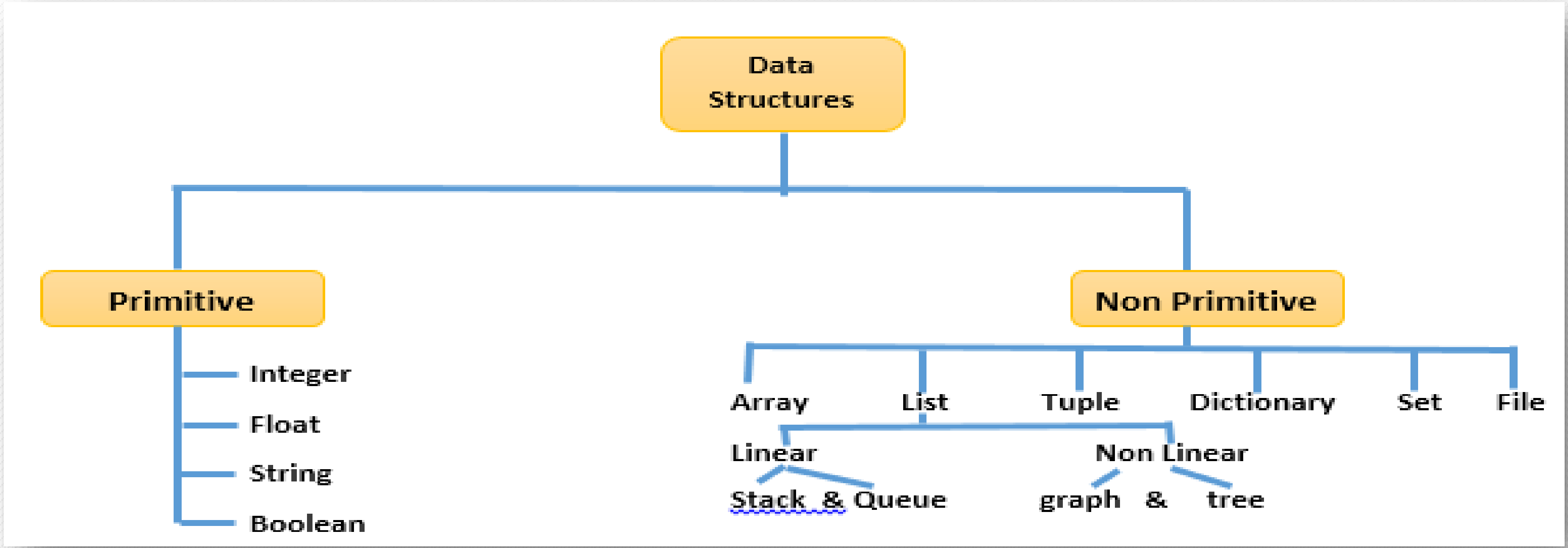
Class XII (As per CBSE Board)

Visit : python.mykvs.in for regular updates

Data-structures

It a way of organizing and storing data in such a manner so that it can be accessed and work over it can be done efficiently and less resources are required. It define the relationship between the data and the operations over those data. There are many various types of data structures defined that make it easier for the computer programmer,to concentrate on the main problems rather than getting lost in the details of data description and access.

Python Data Structure

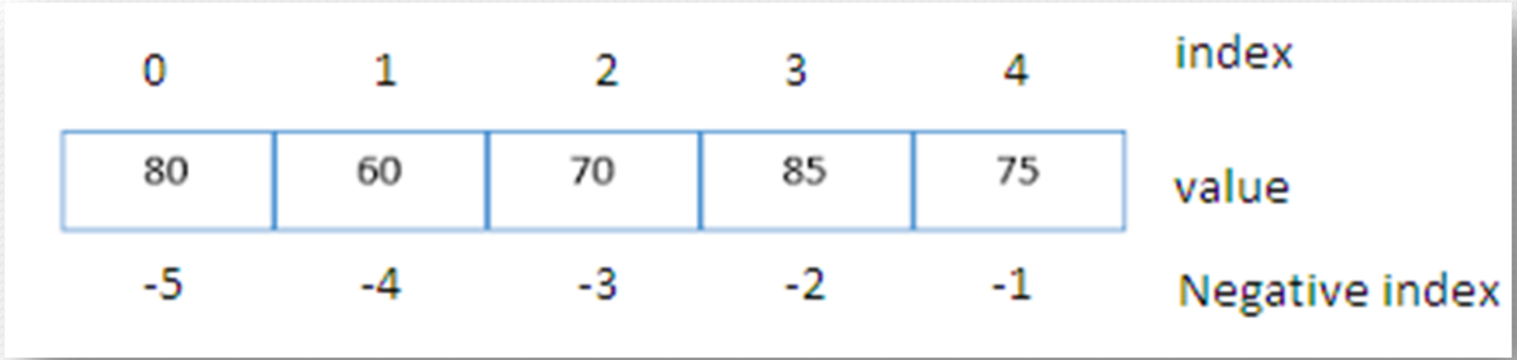


Data-structures

List

It is a collections of items and each item has its own index value.
Index of first item is 0 and the last item is n-1.Here n is number of items in a list.

Indexing of list



Creating a list

Lists are enclosed in square brackets [ ] and each item is separated by a comma.

<u>e.g.</u>

```
list1 = [‘English', ‘Hindi', 1997, 2000];
list2 = [11, 22, 33, 44, 55 ];
list3 = ["a", "b", "c", "d"];
```

Access Items From A List

List items can be accessed using its index position.

e.g.

```
list =[3,5,9]
print(list[0])
print(list[1])
print(list[2])
print('Negative indexing')
print(list[-1])
print(list[-2])
print(list[-3])
```

output

```
3
5
9
Negative indexing
9
5
3
```

Iterating Through A List

List elements can be accessed using looping statement.
e.g.

```
list =[3,5,9]
for i in range(0, len(list)):
    print(list[i])
```

Output

```
3
5
9
```

Data-structures

Important methods and functions of List

Function	Description
list.append()	Add an Item at end of a list
list.extend()	Add multiple Items at end of a list
list.insert()	insert an Item at a defined index
list.remove()	remove an Item from a list
del list[index]	Delete an Item from a list
list.clear()	empty all the list
list.pop()	Remove an Item at a defined index
list.index()	Return index of first matched item
list.sort()	Sort the items of a list in ascending or descending order
list.reverse()	Reverse the items of a list
len(list)	Return total length of the list.
max(list)	Return item with maximum value in the list.
min(list)	Return item with min value in the list.
list(seq)	Converts a tuple, string, set, dictionary into list.

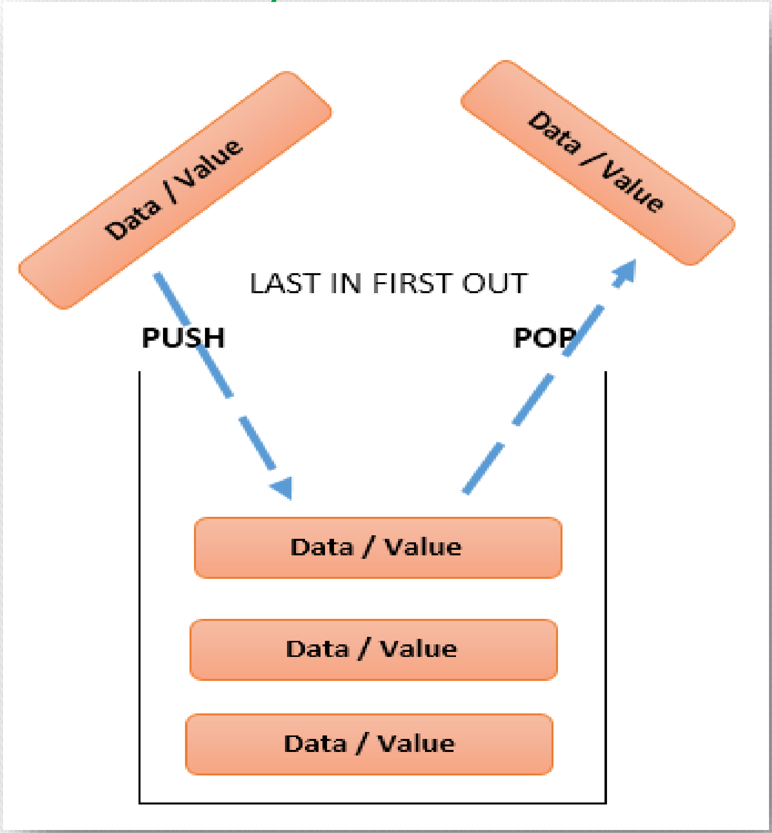
For detail on list click here

Data-structures

Stack:

A stack is a linear data structure in which all the insertion and deletion of data / values are done at one end only.

- It is type of linear data structure.
- It follows LIFO(Last In First Out) property.
- Insertion / Deletion in stack can only be done from top.
- Insertion in stack is also known as a PUSH operation.
- Deletion from stack is also known as POP operation in stack.



Applications of Stack:

- Expression Evaluation: It is used to evaluate prefix, postfix and infix expressions.
- Expression Conversion: It can be used to convert one form of expression(prefix,postfix or infix) to one another.
- Syntax Parsing: Many compilers use a stack for parsing the syntax of expressions.
- Backtracking: It can be used for back traversal of steps in a problem solution.
- Parenthesis Checking: Stack is used to check the proper opening and closing of parenthesis.
- String Reversal: It can be used to reverse a string.
- Function Call: Stack is used to keep information about the active functions or subroutines.

Using List as Stack in Python:

The concept of Stack implementation is easy in Python , because it support inbuilt functions (append() and pop()) for stack implementation.By Using these functions make the code short and simple for stack implementation.

To add an item to the top of the list, i.e., to push an item, we use append() function and to pop out an element we use pop() function. These functions work quiet efficiently and fast in end operations.

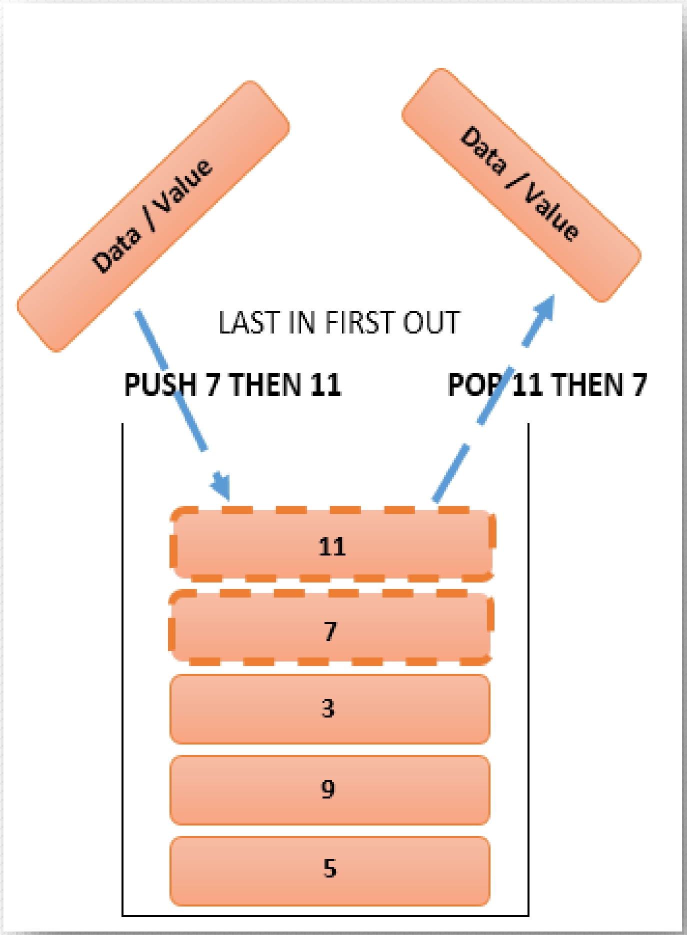
Data-structures

Stack e.g. program:

```
stack = [5, 9, 3]
stack.append(7)
stack.append(11)
print(stack)
print(stack.pop())
print(stack)
print(stack.pop())
print(stack)
```

OUTPUT

```
[5, 9, 3, 7, 11]
11
[5, 9, 3, 7]
7
[5, 9, 3]
```



Data-structures

Stack interactive program:

```
class Stack:
    def __init__(self):
        self.items = []
     def is_empty(self):
        return self.items == []
     def push(self, data):
        self.items.append(data)
    def pop(self):
        return self.items.pop()
s = Stack()
while True:
    print('Press 1 for push')
    print('Press 2 for pop')
    print('Press 3 for quit')
    do = int(input('What would you like to do'))
    if do == 1:
        n=int(input("enter a number to push"))
        s.push(n)
    elif do == 2:
        if s.is_empty():
            print('Stack is empty.')
        else:
            print('Popped value: ', s.pop())
    elif operation == 3:
        break    #Note :- Copy and paste above code in python file then execute
that file
```