# National University of Computer & Emerging Sciences

# Lab Manual

## CS461: Artificial Intelligence Lab

| | |
|---|---|
| Course Instructor | Dr. Hafeez-Ur-Rahman |
| Lab Instructor | Muhammad Yousaf |
| Semester | Spring 2021 |

## What is Artificial Intelligence?

The father of Artificial Intelligence, John McCarthy, defines it as

"The science and engineering of making intelligent machines, especially intelligent computer programs"

Artificial Intelligence (AI) Software is a computer program which mimics human behavior by learning various data patterns and insights.

AI is a fundamental risk to the existence of human civilization (Tech Tycoon Alan Musk).

## Types of Artificial Intelligence?

AI Narrow Intelligence: weak AI, applying AI on specific tasks. Alexa for limited functions, autopilot feature in Tesla, google maps, Face recognition in iPhone etc.

AI General Intelligence: strong AI, involves machines that possess the ability to perform any intellectual task that a human being can (consciousness and mind).

AI Super Intelligence: Super AI, time when the capability of computers will surpass humans in future. Hypothetical like movies and science fiction books etc.

## Top 10 Artificial Intelligence Software's

| AI Tools | Functionality | Supported OS/ Languages/Platform | Best Feature | Price |
|---|---|---|---|---|
| Content DNA Platform | Machine Learning/Computer Vision. | Suits both Cloud and On-premises deployment models. | Unsupervised Machine learning.<br>Training on your data. | Onetime fee. |
| Google Cloud Machine Learning Engine | Machine Learning | GCP Console | Trains model on your data.<br>Deploy it.<br>You can manage it. | Per hour per training unit costs: US: $0.49 |
| Azure Machine Learning Studio | Machine Learning | Browser based | Model will get deployed as a web service. | Free |
| Tensor Flow | Machine Learning | Desktops,<br>Clusters,<br>Mobile, | It is for everyone from beginners to experts. | Free |

| | | | | |
|---|---|---|---|---|
| | | Edge devices, CPUs, GPUs, & TPUs. | | |
| H2O AI | Machine Learning | Distributed in-memory Programming<br>Languages: R & Python. | AutoML functionality included. | Free |
| Cortana | Virtual Assistant | Windows, iOS, Android, and Xbox OS.<br>Supported Languages: English, Portuguese, French, German, Italian and Chinese etc. | It can perform so many tasks from setting reminders to switching on the lights. | Free |
| IBM Watson | Question-answering system. | SUSE Linux Enterprise Server 11 OS Apache Hadoop framework. | It learns lot from small data. | Free |
| Salesforce Einstein | CRM system | Cloud based. | No need for managing models and data preparation. | Contact them for pricing details |
| Infosys Nia | Machine Learning Chatbot. | Supported devices: Windows, Mac, & Web based. | It provides three components, i.e. Data platform, Knowledge platform, and automation platform. | Contact them for pricing details. |
| Amazon Alexa | Virtual Assistant | OS: Fire OS, iOS, & Android.<br>Languages: English, French, German, Japanese, Italian, and Spanish. | It can be connected to devices like Camera, lights, and entertainment systems. | Free with some amazon devices or services. |
| Google Assistant | Virtual Assistant | OS: Android, iOS, and KaiOS.<br>Languages: English, Hindi, German, Italian, Japanese, Spanish etc. | Supports two way conversation. | Free |

Lab # 01

# Data Structures Revision (Python)

## Recursion:

Recall fibonacci series:

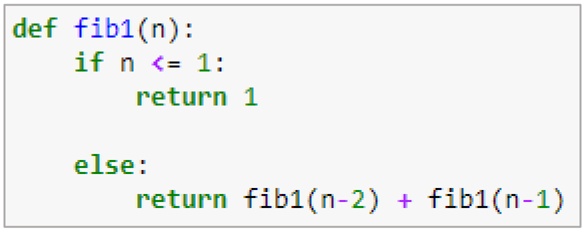
1,1,2,3,5,8,13,...1,1,2,3,5,8,13,...

fib (n) = fib(n−1) + fib(n−2)

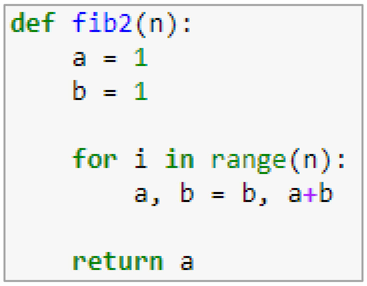
Recall Factorial!

Fact (n) = n*(n-1)*(n-2)*(n-3)*……….*1

We had two alternatives for computing fib (n)



```
def fib1(n):
    if n <= 1:
        return 1

    else:
        return fib1(n-2) + fib1(n-1)
```



```
def fib2(n):
    a = 1
    b = 1

    for i in range(n):
        a, b = b, a+b

    return a
```

## Linked List:

The Push Operation

Push operation has two cases:

1. When there are no nodes
2. When there is already one or more nodes

The Pop Operation

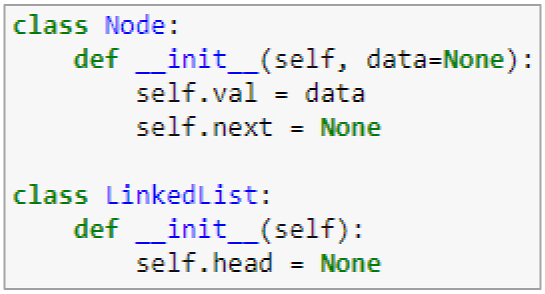
Pop also has two cases:

1. When there is only one node
2. When there are 2 or more nodes -- in this case, we keep two pointers: prev and temp. Move both until temp is the last. Then set next of prev to None

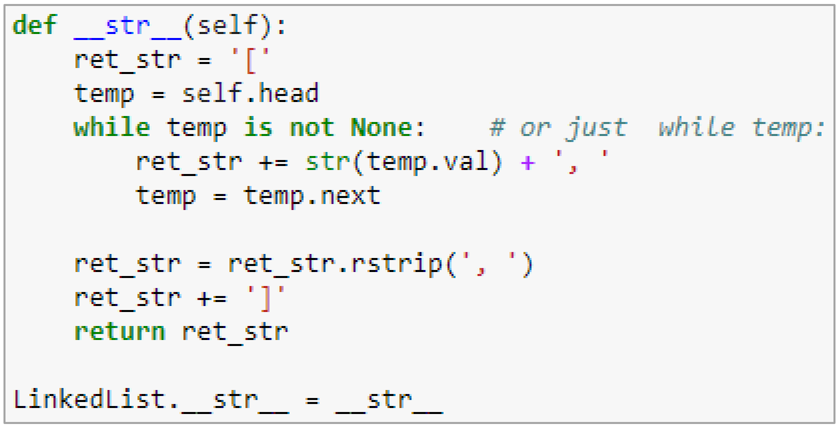
Conversion to String

Python has a special function __str__. This is called whenever a cast to string is made. (These are called dunder (double underscore) functions.)

Lab # 01



```
class Node:
    def __init__(self, data=None):
        self.val = data
        self.next = None

class LinkedList:
    def __init__(self):
        self.head = None
```



```
def __str__(self):
    ret_str = '['
    temp = self.head
    while temp is not None:     # or just  while temp:
        ret_str += str(temp.val) + ', '
        temp = temp.next

    ret_str = ret_str.rstrip(', ')
    ret_str += ']'
    return ret_str

LinkedList.__str__ = __str__
```

Insertion

Again, two cases:

1. Insertion at index 0: new head, old head becomes next of this new head
2. Insertion at any other index: in this case, move prev and temp forward index times. Then, insert new node between prev and temp.

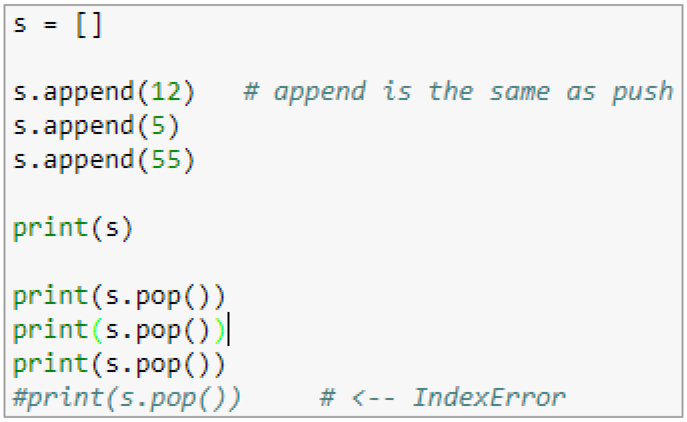
Remove Operation

This is also the same:

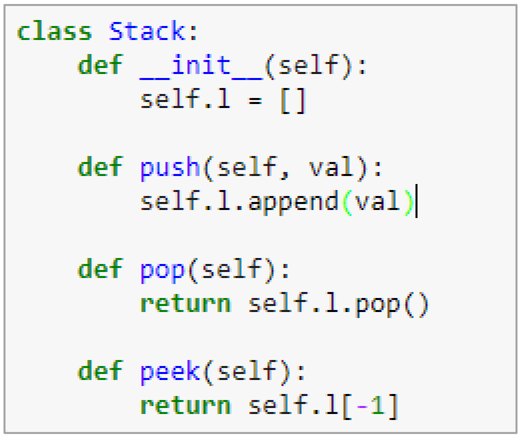
1. If first node is present and same as val, remove it.
2. Otherwise, move prev and temp until temp points to the value. Set next of prev to next of temp. (Temp is lost)

## Stack

A stack is essentially free in Python. Here's what happens when we use Python's list.



```
s = []

s.append(12)   # append is the same as push
s.append(5)
s.append(55)

print(s)

print(s.pop())
print(s.pop())
print(s.pop())
#print(s.pop())     # <-- IndexError
```

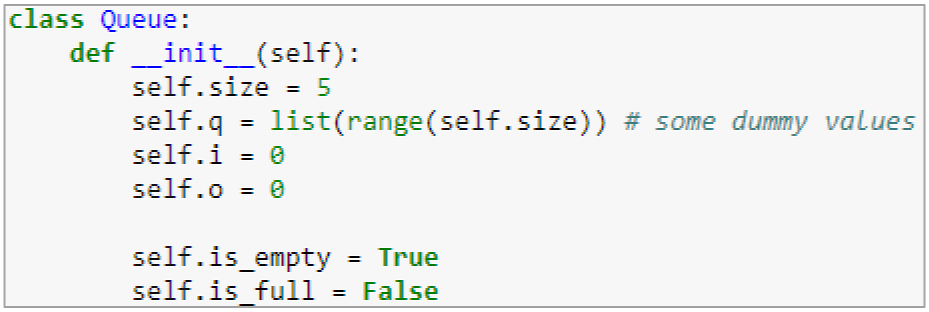


```
class Stack:
    def __init__(self):
        self.l = []

    def push(self, val):
        self.l.append(val)

    def pop(self):
        return self.l.pop()

    def peek(self):
        return self.l[-1]
```

## Queue

Implement queue data structure in python with the following basic functions;

1. __init__(self)
2. __str__(self)
3. inc
4. Enqueue
5. Dequeue



```python
class Queue:
    def __init__(self):
        self.size = 5
        self.q = list(range(self.size)) # some dummy values
        self.i = 0
        self.o = 0

        self.is_empty = True
        self.is_full = False
```