DAILY ASSESSMENT FORMAT

Date:	**22/05/20202**	**Name:**	**Kishan shetty**
Course:	**TCS-ION**	**USN:**	**4AL17ec041**
Topic:	**Understand Artificial Intelligence (AI) - Part 1** **Understand Artificial Intelligence (AI) - Part 2**	**Semester & Section:**	**6th,A**
Github Repository:	**Kishanshetty-041**		

FORENOON SESSION DETAILS

TCS iON | Digital Learning Hub
Learn, Share, Collaborate

TATA CONSULTANCY SERVICES

This is to certify that

Kishan Shetty

has successfully completed

Career Edge - Knockdown the Lockdown

online course offered by TCS iON

Start Date: 17 May 2020 | **End Date:** 22 May 2020

Topics:

- Communication Skills
- Presentation Skills
- Soft Skills
- Career Guidance Framework
- Resume Writing
- Group Discussion Skills
- Interview Skills
- Business Etiquette
- Effective Email Writing
- Telephone Etiquette
- Accounting Fundamentals
- IT Foundational Skills
- Overview of Artificial Intelligence* (Source: NPTEL)

CERTIFIED
Cert. ID.: 4-8316286-1016
Dated: 22 May 2020

Mehul Mehta
Mehul Mehta
Global Delivery Head, TCS iON

Report –

Understanding Artificial Intelligence

Goals of Artificial Intelligence:

- **Introduce you to the Yield of AI**
- **To Explain the Challenges in Building on Intelligent System**
- **To Explain the**
  - ❑ **Key Paradigms**
  - ❑ **Core Techniques**
  - ❑ **Algorithms**
- **After this Course you will be able to**
  - ❑ **Formulate Problems as State Space Search,Problems and Efficiently Solve Them**
  - ❑ **Write Game Playing Programs**
  - ❑ **Use Machine Learning to Find Patterns to Data**
  - ❑ **Building Expert Systems**

What is Intelligence ?

- **Behave as Intelligently as Humans**
- **Behave in the Best Possible Manner**
- **Thinking**
- **Acting**

Typical AI Problems :

- **Intelligent Entities need to be able to do both "Mundane "and" Expert" Tasks.**
  - ❑ **Planning Route,Activity**
  - ❑ **Recognizing People,Objects**
  - ❑ **Communicating**
  - ❑ **Navigating Around Obstacles on the Street**
- **Expert Tasks :**
  - ❑ **Medical Diagnosis**
  - ❑ **Mathematical Problem Solving**

Intelligence Behaviour

- **Perception**
- **Reasoning**
- **learning**
- **Understanding Language**
- **Solving Problems**

Applications:

- **Computer Vision**
- **Image Recognition**
- **Robotics**
- **Language Processing**

- **Speech Processing**

Internet Agents :

- **Monitor user Tasks**
- **Seeks Needed Information**
- **Learn Which Information is Most Useful**

AI Topics

- **Core Areas**
- **Perception**
- **Uncertainty**
- **General Algorithms**
- **Applications**
- **Decision Theory Reasoning with Symbolic Data**

Limits of AI Today

- **Today Successful AI System**
- ❑ **Operate in well-Defined Domains**
- ❑ **Employ Narrow,Specialized Knowledge**
- **Commonsense Knowledge**
- ❑ **Needed in Complex,Opens Ended Worlds**
- ❑ **Understand Unconstrained Natural Language**

What can AI Systems do ?

- **Computer Vision :Face Recognition**
- **Robotics : Autonomous Automobile**
- **Natural Language Processing :Simple Machine Translation**
- **Expert Systems : Medical Diagnosis in a Narrow Domain**
- **Spoken Language : 1000 Words Continuous Speech**
- **Planning and Scheduling : Hubble Telescope Experiments**
- **Learning**
- **Gaming : Grand Master Level in Chess(World Champion),Checkers,etc..**

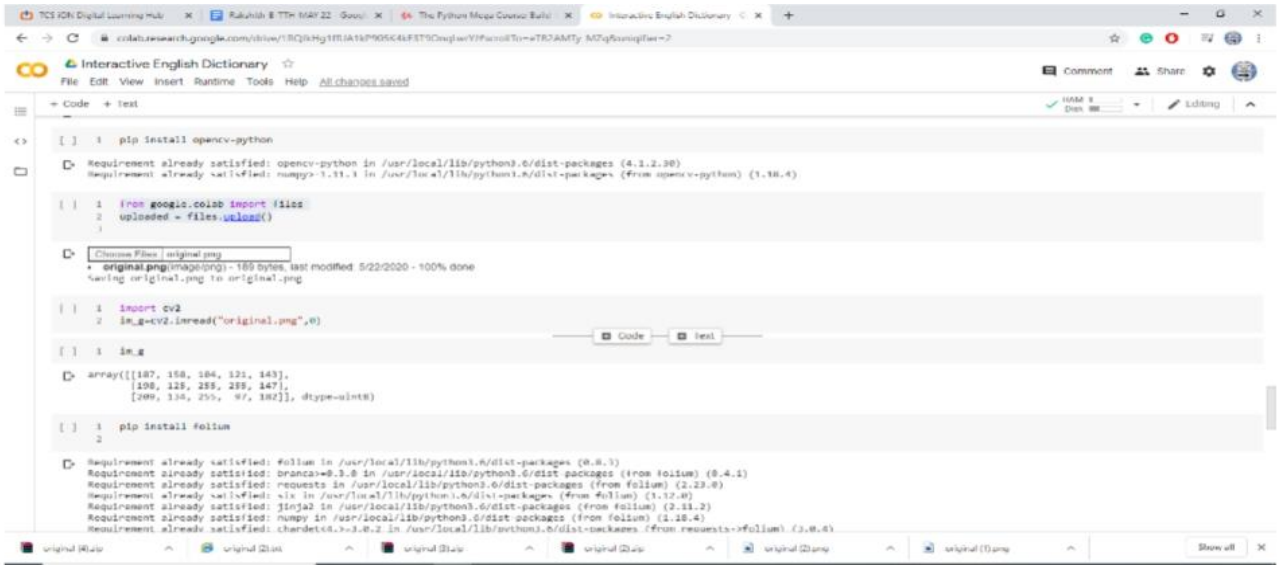
What AI Cannot do ?

- **Understand Natural Language Robustly**
- **Read and Understand article in a Newspaper**
- **Surf the Web**
- **Learn a Natural Language**

Date:	**22/05/2020**	**Name:**	**Kishan shetty**
Course:	**Python**	**USN:**	**4AL17EC041**
Topic:	**Application 2: Create Webmaps with Python and Folium**	**Semester & Section:**	**6th,A**

AFTERNOON SESSION DETAILS

Image of session



Report –

```
import folium
import pandas

data = pandas.read_csv("Volcanoes.txt")
lat = list(data["LAT"])
lon = list(data["LON"])
elev = list(data["ELEV"])

def color_producer(elevation):
   if elevation < 1000:
      return 'green'
   elif 1000 <= elevation < 3000:
      return 'orange'
   else:
      return 'red'
map = folium.Map(location=[38.58, -99.09], zoom_start=6, tiles="Mapbox Bright")

fgv = folium.FeatureGroup(name="Volcanoes")

for lt, ln, el in zip(lat, lon, elev):
   fgv.add_child(folium.CircleMarker(location=[lt, ln], radius = 6, popup=str(el)+" m",
   fill_color=color_producer(el), fill=True,  color = 'grey', fill_opacity=0.7))

fgp = folium.FeatureGroup(name="Population")

fgp.add_child(folium.GeoJson(data=open('world.json', 'r', encoding='utf-8-sig').read(),
style_function=lambda x: {'fillColor':'green' if x['properties']['POP2005'] < 10000000
else 'orange' if 10000000 <= x['properties']['POP2005'] < 20000000 else 'red'}))

map.add_child(fgv)
map.add_child(fgp)
map.add_child(folium.LayerControl())

map.save("Map1.html")
```