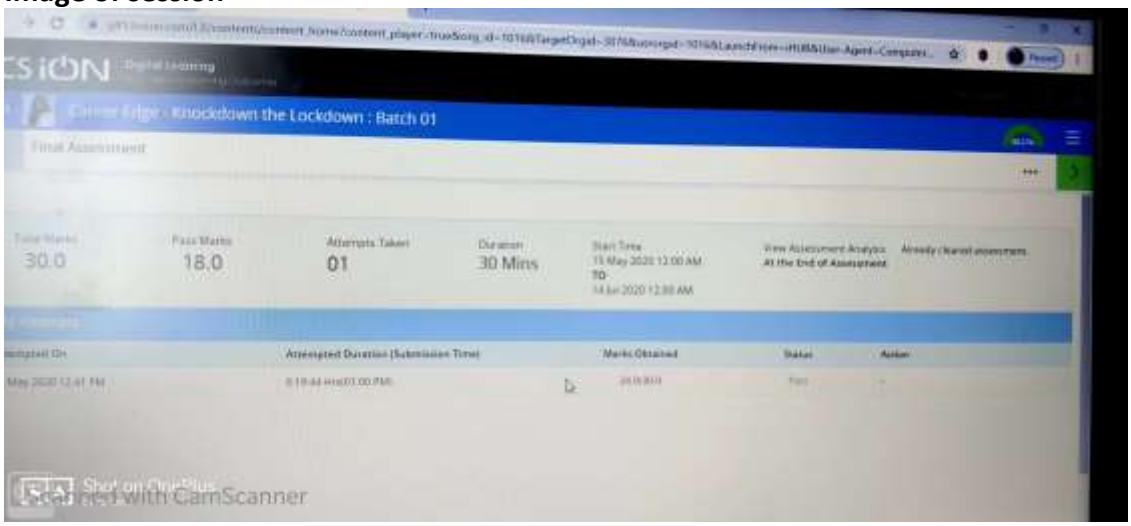
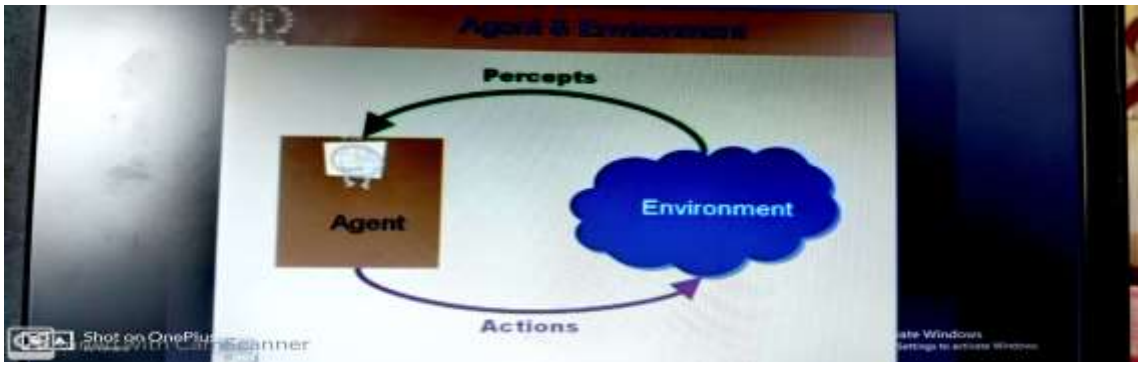
DAILY ASSESSMENT FORMAT

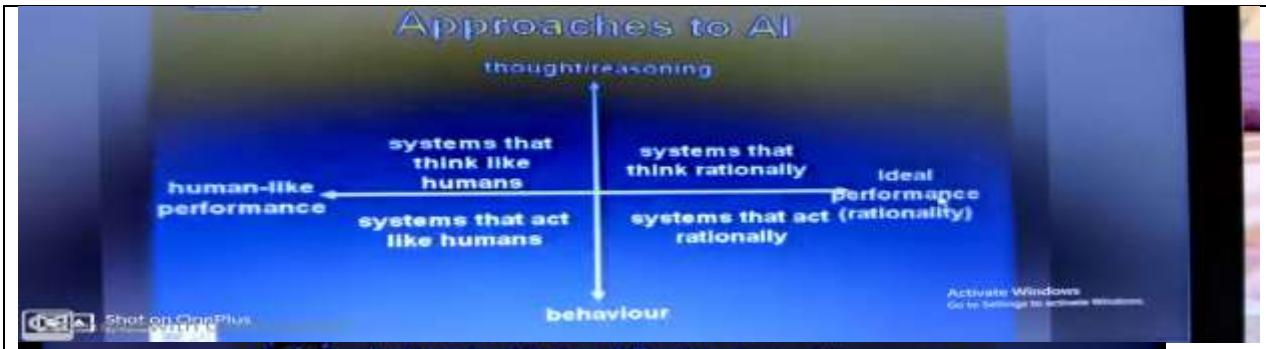
Date:	**22 may 2020**	**Name:**	**Yamunashree N**
Course:	**TCS ION CAREER EDGE**	**USN:**	**4AL17EC097**
Topic:	**Understand Artificial Intelligence(AI)-part 1 Understand Artificial Intelligence(AI)-part 2 Assessment**	**Semester & Section:**	**6th sem B sec**
Github Repository:	**yamunashree-courses**		

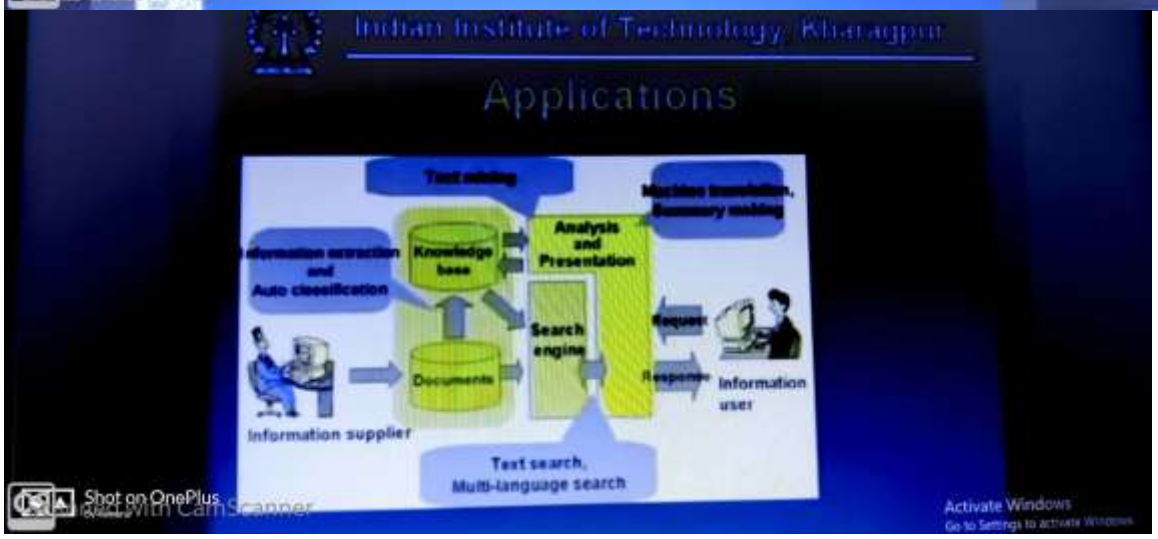
FORENOON SESSION DETAILS

Image of session






Approaches to AI
thought/reasoning
systems that think like humans
systems that think rationally
human-like performance
ideal performance (rationality)
systems that act like humans
systems that act rationally
behaviour
Activate Windows


Indian Institute of Technology, Kharagpur
Applications
Knowledge base
Analysis and Presentation
Search engine
Documents
Information supplier
Information user
Request
Response
Test search, Multi-language search
Activate Windows

Artificial Intelligence (Part 1) Day5

Introduction:

What is AI?
=> is concerned with the design of intelligence in artificial device.

Typical AI problems
-> Intelligent entities (agents) need to be able to do both "mundane" and "expert" tasks:

Mundane tasks:-
-> Planning routes, activity.
-> Recognizing (through vision) people, objects.
-> communicating (through natural language).
-> Navigating around obstacles on the street

Expert tasks:-
-> Medical diagnosis
-> Mathematical problem solving

AI systems can easily do symbolic integration, proving theorems playing chess, internal diagnosis

Intelligent Behaviour
-> Perception
-> Reasoning
-> Learning
-> Understanding Language
-> Problem-Solving Problem

Scanned with CamScanner

Applications
-> Computer Vision
-> image recognition
-> Robotics
-> Language understanding

Autonomous Land Vehicle in a Neural Network :-
-> 1959 - Dean Pomerleau at CMU creates ALVINN

Deep Blue - 1997 chess program beats the current world class champion, Gary Kasparov in a widely follo match

-> Machine Translation
- Carnegie Mellon is working on its own 'Speechlat for use in doctor-patient interviews.

Approaches to AI:-

Strong AI aims to build machines that can truly reason and solve problems which is a self aware and whose overall intellectual ability is indistingu ishable from that of a human being
-> human-like
-> Non-human like

Weak AI :- deals with creation of some form of computa based artificial intelligence that cannot truly reason and solve problems, but can act as it were intelligent

Strong AI : maintains that suitably programmed machines are capable of cognitive mental state

Applied AI : aims to produce commercially viable Systems

Cognitive AI :- Computers are used to test theories about how the human mind works.

Artificial Intelligence Part 2 :-

Sensor & Effectors :-

An agent perceives its environment through sensors

-> the complete set of inputs at a given time is called a percept.

-> The current percept or a sequence of percepts can influence the actions of an agent

It can change the environment through effectors

-> An operation involving an actuator is called an action

-> Actions can be grouped into action sequences

Agents as Sensors, Actuators

Have goals

-> implement mapping from percept sequence to action

Examples of Agents :-

Human :- Ears, Eyes, Taste buds etc.

Robot - Camera, bumper, infrared etc

Software ~~aged~~ agent :- softbot

Episodic | Sequential :-

An episodic environment means that subsequent episodes do not depend on what actions occurred in previous episodes.

-> In a sequential environment the agent engages in a series of connected episodes.

Dynamism -

Static -> does not change when agent is delivered

Dynamic -> Changes by itself apart from the action agent takes.

Discrete -> If the number of distinct percepts and actions is limited, the environment is discrete. otherwise it is continuous.

Knowledge rich environment :- enormous amount of information that the environment contains.

Input rich :- the enormous amount of input the environment can send to an agent T.

mapping is implicitly defined by a program

-> rule based

-> neural networks

-> algorithm

Subsumption Architecture

Rodeny Brooks - 1988

Sensory input - Action (lower animals)

Features

-> No explicit knowledge representation

-> Distributed behaviour

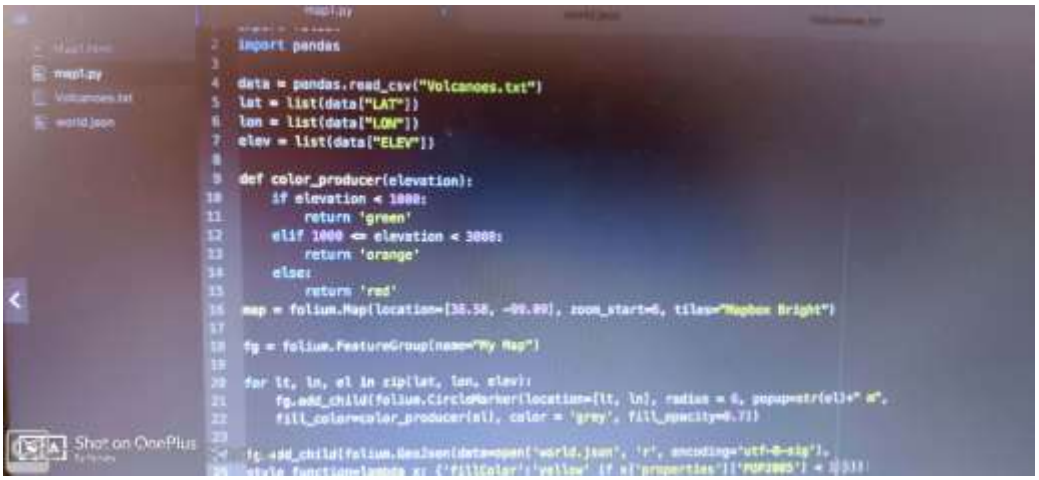
-> Response to stimuli is reflexive

Interpreting individual agents

Date:	22 may 2020	Name:	Yamunashree N
Course:	Python Programming	USN:	4AL17EC097
Topic:	Application 2: create webmaps with python and folium	Semester & Section:	6th sem B sec

AFTERNOON SESSION DETAILS

Image of session



Application 2: Create Webmaps with Day 5
Python and Folium

1. install folium
(jinja2 -> folium)
2. install jinja2
3. python

Terminal

Create basic Webmap
-> import folium
-> map = folium.Map / dir(folium)
-> help(folium.Map)
map = folium.Map(location=[80, -100])
-> map
-> map.save("Map1.html")

titles = "Stamen Terrain" titles = "Mapbox Bright"
are both types of base maps, but Mapbox Bright
doesn't work anymore Stamen Terrian work great
and you will see it creates a Beautiful relief map

Adding Points:-
-> help(folium.Map)
import folium
map = folium.Map(location=[38.58, -99.09], zoom_
start=6, tiles="Mapbox Bright")
fg = folium.FeatureGroup(name="My Map")
fg.add_child(folium.Marker(location=[38.2, -99.
"Marker", icon=folium.Icon

Picker='green')))
map.add_child(fg)
map.save("Map1.html")

Adding Multiple points:-

import folium
map = folium.Map(location=[38.58, -99.09], zoom_
start=6, tiles="Mapbox Bright")
fg = folium.FeatureGroup(name="My Map")
for coordinate in [[38.2, -99.1], [39.2, -97.1]]:
fg.add_child(folium.Marker(location=coordinate
popup="Hi I am a Marker", icon=folium.Icon(color
='green'))
map.add_child(fg)
map.save("Map1.html")

Adding points from file:-

import folium
import pandas
data = pandas.read_csv("Volcanoes.txt")
lat = list(data["LAT"])
lon = list(data["LON"])
map = folium.Map(location=[38.58, -99.09], zoom_
start=6, tiles="Mapbox Bright")
fg = folium.FeatureGroup(name="My Map")
for lt, ln in zip(lat, lon):
fg.add_child(folium.Marker(location=[lt, ln]
popup="Hi I am a Marker", icon=folium.Icon

color = 'green'))
map.add_child(fg)
map.save("Map1.html")

Pop Windows on Map

```
elev = list(data["ELEV"])
```

Color Points :-

```
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")
fg = folium.FeatureGroup(name="My Map")
for lt, ln, el in zip(lat, lon, elev):
    fg.add_child(folium.Marker(location=[lt, ln],
    popups=str(el)+"m", icon=folium.Icon(color=
                color_producer(el))))
map.add_child(fg)
map.save("Map1.html")
```