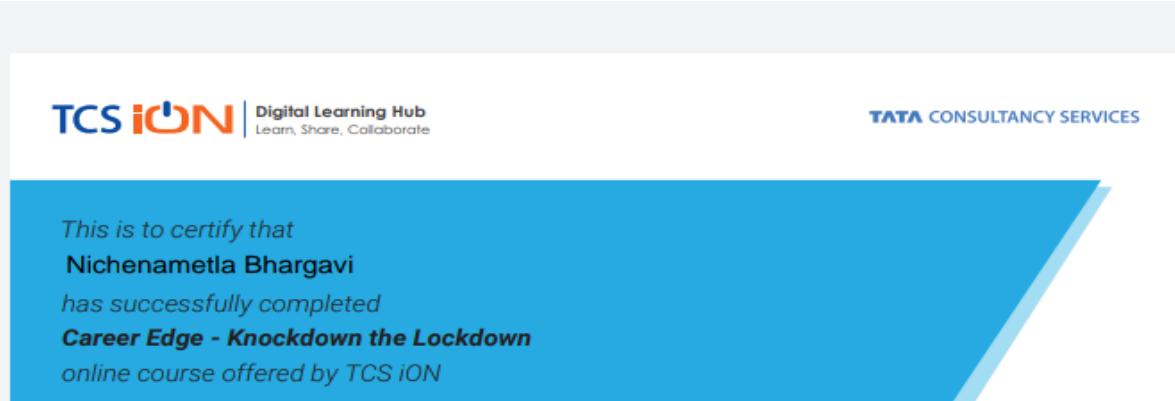


DAILY ASSESSMENT FORMAT

Date:	22/5/2020	Name:	Nichenametla Bhargavi
Course:	TCS-ION CAREER EDGE	USN:	4AL17EC061
Topic:	Understand Artificial Intelligence - Part-1, Part-2, Assessment	Semester & Section:	6th Sem A Sec
Github Repository:	alvas-education-foundation/Bhargavi_Nichenametla		

FORENOON SESSION DETAILS																											
<p>Image of session</p>  <table border="1" style="width: 100%; border-collapse: collapse; margin-top: 10px;"> <thead> <tr> <th colspan="2">My Attempts</th> <th colspan="3"></th> <th colspan="2"></th> </tr> <tr> <th>Attempted On</th> <th>Attempted Duration (Submission Time)</th> <th>Marks Obtained</th> <th>Status</th> <th>Action</th> <th colspan="2"></th> </tr> </thead> <tbody> <tr> <td>22 May 2020 12:48 PM</td> <td>0:15:25 Hrs(01:03 PM)</td> <td>28.0/30.0</td> <td>Pass</td> <td>-</td> <td colspan="2"></td> </tr> </tbody> </table>  <p style="text-align: center; font-size: small;"> 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) </p> <p style="text-align: left; margin-top: 20px;">  Cert. ID: 4-8314600-1016 Dated: 22 May 2020 </p> <p style="text-align: right; margin-top: 20px;">  Mehul Mehta Global Delivery Head, TCS iON </p>							My Attempts							Attempted On	Attempted Duration (Submission Time)	Marks Obtained	Status	Action			22 May 2020 12:48 PM	0:15:25 Hrs(01:03 PM)	28.0/30.0	Pass	-		
My Attempts																											
Attempted On	Attempted Duration (Submission Time)	Marks Obtained	Status	Action																							
22 May 2020 12:48 PM	0:15:25 Hrs(01:03 PM)	28.0/30.0	Pass	-																							

Report – Report can be typed or hand written for up to two pages.

Autonomous agents

- In space exploration, robotic space probes autonomously monitor their surroundings, make decisions and act to achieve their goals.

Mars Exploration Rover Mission

[Mars Exploration Rover Mission Home.htm](#)

Machine translation

- Immediate translations between people speaking different languages would be a remarkable achievement of enormous economic and cultural benefit.
- Universal translation is one of 10 emerging technologies that will affect our lives and work 'in revolutionary ways' within a decade, Technology Review says.

Typical AI Problems

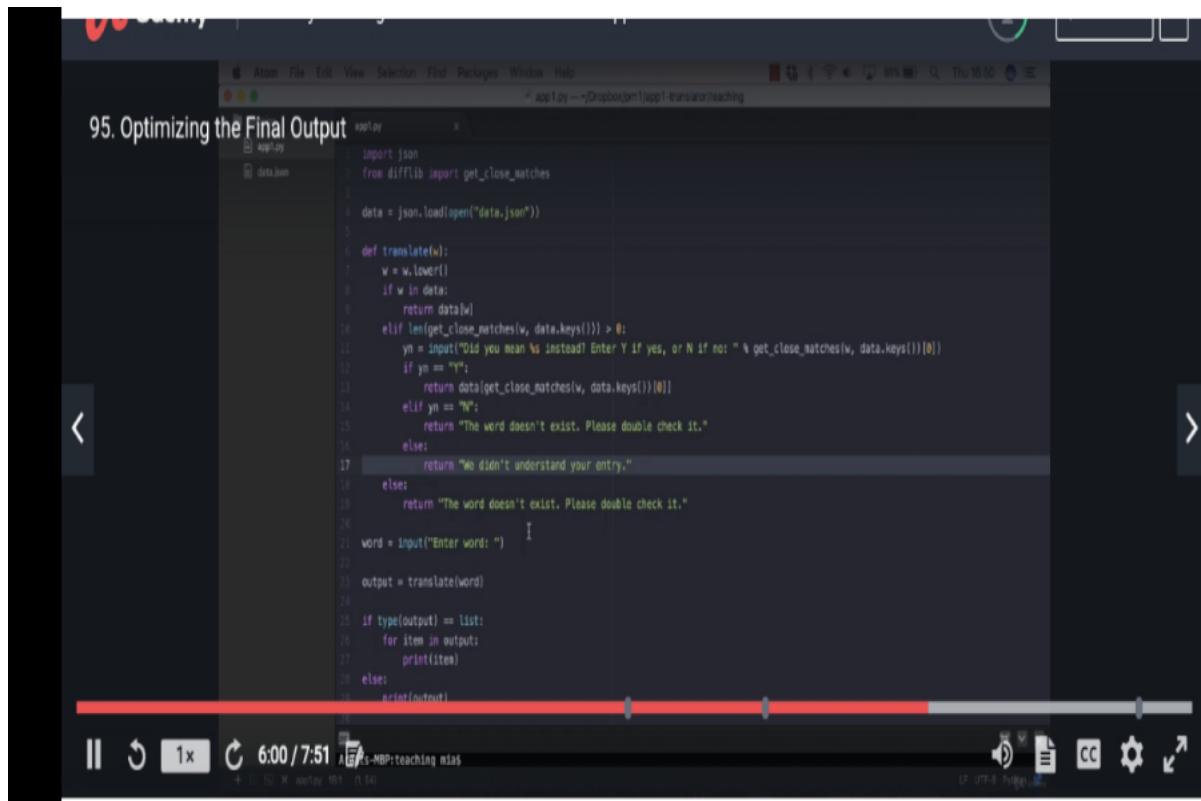
- Intelligent entities (or "agents") need to be able to do both "mundane" and "expert" tasks:
- Mundane tasks :
 - Planning route, activity.
 - Recognizing (through *vision*) people, objects.
 - Communicating (through *natural language*).
 - Navigating round obstacles on the street
- Expert tasks :

[Medical diagnosis](#)

Date:	22/05/2020	Name:	Nichenametla Bhargavi
Course:	Python Bootcamp 2020 build 15 working applications and Games	USN:	4AL17EC061
Topic:	Application 1:Build an Interactive English dictionary	Semester & Section:	6th Sem A sec

AFTERNOON SESSION DETAILS

Image of session



The screenshot shows a dark-themed Atom code editor window titled "95. Optimizing the Final Output". The code is written in Python and performs word translation and handling of close matches. It includes imports for json and difflib, loads data from a JSON file, defines a translate function, and handles user input for close matches.

```

95. Optimizing the Final Output
  import json
  from difflib import get_close_matches

  data = json.load(open("data.json"))

  def translate(w):
    w = w.lower()
    if w in data:
      return data[w]
    elif len(get_close_matches(w, data.keys())) > 0:
      yn = input("Did you mean %s instead? Enter Y if yes, or N if no: " % get_close_matches(w, data.keys())[0])
      if yn == "Y":
        return data[get_close_matches(w, data.keys())[0]]
      elif yn == "N":
        return "The word doesn't exist. Please double check it."
      else:
        return "We didn't understand your entry."
    else:
      return "The word doesn't exist. Please double check it."

  word = input("Enter word: ")
  output = translate(word)

  if type(output) == list:
    for item in output:
      print(item)
  else:
    print(output)

```

Report – Report can be typed or hand written for up to two pages.

```
>>>import json  
>>>data = json.load(open("file location"))  
>>>data ("rain")  
→ dp  
Returning the definition of a word:  
import json  
data = json.load(open("data.json"))  
def translate(word):  
    return data[word]  
word = input("Enter word!")  
print(translate(word))  
If the given word doesn't exist then  
def translate(w):  
    if w in data:  
        return data[w]  
    else:  
        return "The word doesn't exist."  
Implementing Case Sensitivity:  
def translate(w):  
    w=w.lower()  
    if ... - - -
```

```
>>>import difflib  
>>>from difflib import SequenceMatcher  
>>>SequenceMatcher(None, "Rainy", "rain").ratio()  
↳ used for getting similarity of strings.  
>>>from difflib import get_close_matches  
>>>help(get_close_matches)  
get_close_matches(word, possibilities, n=3, cutoff=0.6)  
Best match:  
if w in data:  
    return data[w]  
elif len(get_close_matches(w, data.keys())) > 0:  
    return "Did you mean %s instead?" % get_close_matches(w, data.keys())[0]  
else:  
    return "The word doesn't exist."
```

