DAILY ASSESSMENT FORMAT

Date:	27 may 2020	Name:	karegowda kn
Course:	TCS ION	USN:	4al16ec29
Topic:	Artificial intelligence	Semester & Section:	6th A section
Github Repository:	karegowda courses		

FORENOON SESSION DETAILS
Image of session

12:25

Digital Learning - Career Edge - Knoc...

30.0

Pass Marks

18.0

Attempts Taken

01

Duration

30 Mins

Start Time

17 May 2020 12:00 AM

TO

16 Jul 2020 12:00 AM

View Assessment Analysis

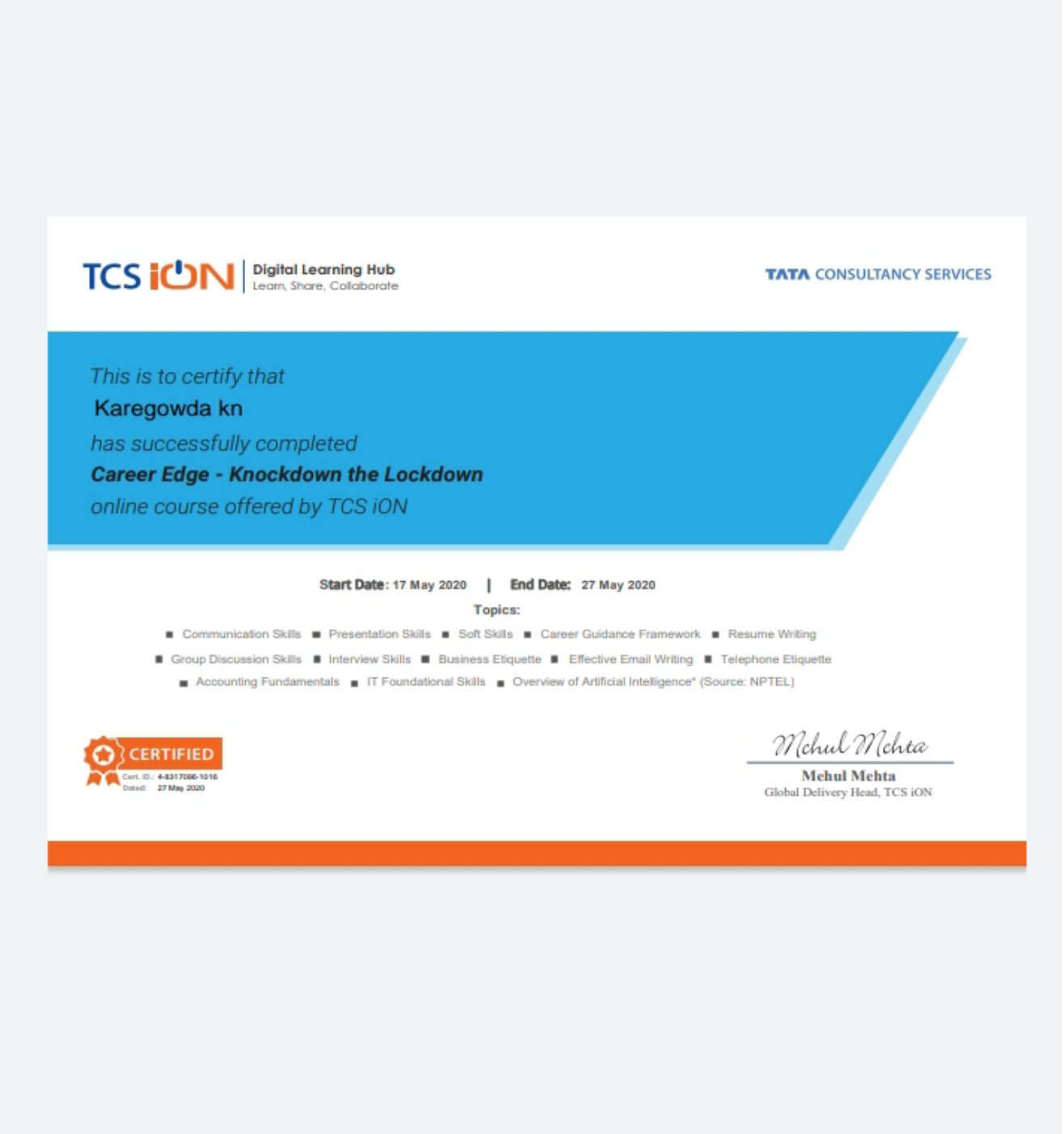
At the End of Assessment

Already cleared assessment.

My Attempts

Attempted On	27 May 2020 11:42 AM
Attempted Duration	0:37:42 Hrs(12:20 PM)
Marks Obtained	22.0/30.0
Status	Pass
Action	-

Cerficate:



TCS iON | Digital Learning Hub
Learn, Share, Collaborate

TATA CONSULTANCY SERVICES

This is to certify that

Karegowda kn

has successfully completed

Career Edge - Knockdown the Lockdown

online course offered by TCS iON

Start Date: 17 May 2020 | **End Date:** 27 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.: 4-8317096-1016
Dated: 27 May 2020

Mehul Mehta

Mehul Mehta
Global Delivery Head, TCS iON

Report--

Instructional objectives:

On completion of this lesson the student will be familiar with:

* Different agent Architectures
* Stimulus response agents
* state based agent
* Deliberative/ goal-directed agents
* Utility based agents
* Learning agents

Sensors and effectors:

* An agent perceives it's environment through sensors
    1. The complete set of input at a given time is called percept
    2. The current percepts or sequence of percepts can influence the actions of agent
* It can change the environment through effectors
    1. An operation involving an actuator is called an action
    2. action can be grouped into action sequence

Types of agents:

* Soft bots
* Expert systems
* Autonomous spacecraft
* Intelligent buildings

Agents:

* Fundamental faculties of intelligence
* In order to act you must sense. blind actions is not a characterisation of intelligence.
* Robotics: sensing and acting, understanding not necessary.
* Sensing needs understanding to be useful.

Rationality:

Rational action: The action that maximizes the expected value of the performance measure given the percept sequence to date.

Environment determinism:

* Deterministic

* Stochastic

* Strategic

Table based agents:

* Information comes from sensors--percepts

* Look it up

* Triggers actions to the effectors

Subsumption Architecture:

* Rodney brooks, 1986

* Sensory input-action

* Brooks -- follow the evolutionary path and simple agents for complex world's

Summary:

* An agent program maps from percept to action and updates its internal state

* Representing knowledge is important for successful agent design

* The most challenging environments are partially observable, stochastic, sequential, dynamic, and continues, and contain multiple intelligence agents.

Day:	26may 2020	Name:	Karegowda kn
Course:	Python	USN:	4al6ec029
Topic:	Putting the pieces together: building a program	Semester & Section:	6th A section

AFTERNOON SESSION DETAILS

Report –

1.pronlem statement:

Here they explain about problem statement there is no code here.

2.approaching the problem

Here they briefly explain how to solve it.

3 building the maker function

```
def sentence_maker(phrase):
   interrogatives = ("why","how","what")
   capitalized = phrase.capitalize()
   if phrase.startswith(interrogatives):
      return "{}?".format(capitalized)
   else:
      return "{}.".format(capitalized)

print (sentence_maker("how are u"))
```

4.constructing the loop

```
def sentence_maker(phrase):
   interrogatives = ("why","how","what")
   capitalized = phrase.capitalize()
   if phrase.startswith(interrogatives):
      return "{}?".format(capitalized)
   else:
      return "{}.".format(capitalized)
```

```
results = [ ]
while True:
    user_input = input("say something:")
    if user_input == "\end":
        break
    else:
        results.append(sentence_maker(user_input))

print(results)
```

5.making the output user-friendly

```
def sentence_maker(phrase):
    interrogatives = ("why","how","what")
    capitalized = phrase.capitalize()
    if phrase.startswith(interrogatives):
        return "{}?".format(capitalized)
    else:
        return "{}.".format(capitalized)

results = []
while True:
    user_input = input("say something:")
    if user_input == "\end":
        break
    else:
        results.append(sentence_maker(user_input))

print(" ".join(results))
```