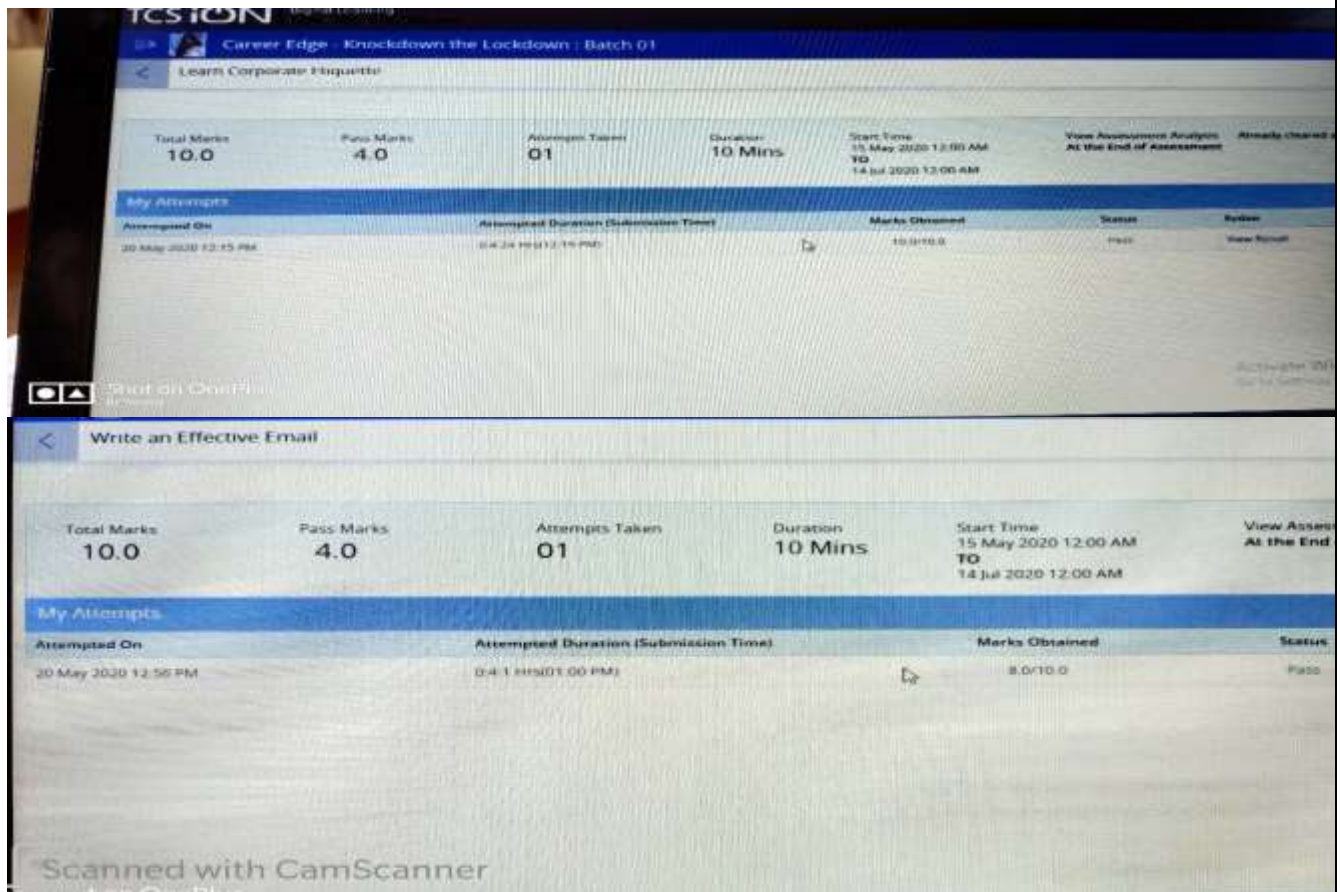


DAILY ASSESSMENT FORMAT

Date:	20 may 2020	Name:	YAMUNASHREE N
Course:	TCS ION CAREER EDGE	USN:	4AL17EC097
Topic:	ACE CORPORATE INTERVIEWS , LEARN CORPORATE ETIQUETTE, WRITE EFFECTIVE Emails	Semester & Section:	6 th sem B sec
Github Repository:	yamunashree-courses		

FORENOON SESSION DETAILS

Image of session



Day 3

Office Corporate Interviews :-

Interview is a skill. Requires Practice and Preparation.

Interview is a widely used process of screening applicants for jobs.

Gary Dessler "An interview is a procedure designed to obtain information from a person's oral response to oral inquiries".

Bovee and Thrall, "An interview is any planned conversation with a specific purpose involving two or more people".

Preparation for a Job interview

- interview preparation includes 4 P's: Prepare, Practice, Present, Participate.
- initial preparation requires a thorough investigation skills, accomplishments, expertise and interests.

Preparation for an effective interview includes :-

- A good assessment of yourself.
- Researching the organization.
- updating your resume.
- Preparing a list of commonly asked questions.

Negotiation Skills :-

- Negotiation is all about a win-win situation.

Techniques

- Do not stick to specific point of negotiation.
- follow different styles and mannerisms to negotiate.

- Be comfortable in whatever style you choose.
- Be calm, relaxed and confident.

Lean Corporate Etiquette :-

Business etiquette can be defined as all the rules that one has to follow when in a business environment.

A firm handshake leaves a positive impression on the person you are communicating with.

Basic Rules:-

- Business cards are used to build your contact list.
- Building network. Do it with poise and confidence.
- In case you don't have your business cards show your contact details like your phone number & email.

Space

Introduction

In a meeting

Language

Volume of voice

Personal Details

Cubicle Etiquette

Do's

- neat and clean
- maintain a calendar
- few decorative items
- Pen & notepad
- Switch off lights and laptop
- Remove post-its

Personal hygiene

Cafeteria

Interrupting a person

Don't's

- Do not talk loudly
- Do not peep
- Do not eat at the work station
- Do not use the office phone for personal use
- Do not loiter around
- Do not play games in your cubicle

Internet Etiquette :-

- Responds to emails online
- Go through FAQs of a site before posing any questions
- Make sure the mail is crisp and has a good subject time
- Send individual mails rather than to an entire group

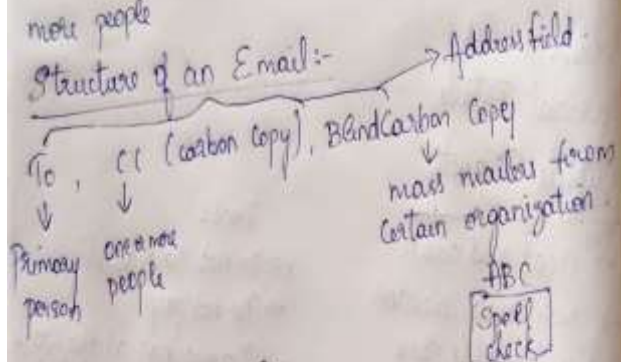
Meeting Etiquette

- Read about the agenda/sub before attending the meeting
- Carry a pen and notepad to the meeting
- Be punctual
- Be attentive and take notes
- Dress appropriately
- Be polite when asking questions

3) Write Effective Emails :-

Email is the short form for Electronic Mail. It is the information sent electronically between two or more people

Structure of an Email :-



Subject :- Single line

Salutation - Hi/Hello etc

Text → 3-4 lines (use bullets)

Complimentary Close :- Thank you etc

Signature

Do's

- Use a strong line
- Type the correct email id
- Keep your email short
- Write ~~email~~ ^{about} only one topic in email
- Be polite and courteous in tone
- Use correct spelling and grammar
- Reply within a reasonable time

Don't's

- Don't use all UPPERCASE
- Don't use unnecessary humor
- Don't use too many short forms, SMS language
- Don't use one word responses
- Don't use email for gossip

Conclusion :- Summary :-

- To write an email using the structure of an email
- To write an effective subject line and text
- To use a few opening and closing phrases
- To use correct spelling and grammar
- Follow company's Email policy
- The importance of responding to emails promptly

Understand How to Ace Corporate Interviews

Where do you see yourself five years down the line?

Make sure your answer:

- Is a blue-print of your aim
- Follow the SMART approach
- Is realistic
- Is not a threat to the interviewer's position

Specific

Measurable

Attainable

Relevant

Timely

Shot on OnePlus

Why should we hire you?

- Talk about your best qualification and experience.
- Talk about how you can be an asset to the company.
- Talk how you can produce exceptional results.
- Talk about your passion for the job profile.
- Talk about your communication skills and abilities to convince.
- Talk about how you can be a team player.

Shot on OnePlus

Understand How to Ace Corporate Interviews

Tell me something about yourself!

- Start with a greeting, for instance: Good morning or Hello everyone!, depending on whom you are speaking to.
- Then introduce yourself by saying "My name is _____ or I am _____ (your name)".
- Never say myself _____ (name).
- Talk about your academic qualifications, for e.g. I have completed my Bachelors Degree In Science from Maharashtra University.
- If you have achieved a First class or distinction, please mention it. If you have higher academic achievements after bachelors, then mention it first.
- After that speak about your technical skills, if you have any special skills, awards, and any certification(s) then you may talk about those.
- If you have work experience, draft a career summary and mention it here.

Shot on OnePlus

Date:	20 may 2020	Name:	YAMUNASHREE N
Course:	PYTHON PROGRAMMING	USN:	4AL17EC097
Topic:	Application 1: Build an interactive English dictionary	Semester & Section:	6 th SEM B SEC

AFTERNOON SESSION DETAILS

Application 1: Build an interactive English Dictionary

Loading JSON Data

```
import json
-> help(json.load)
-> data = json.load(open("data.json"))
-> type(data)
-> data["rain"]
```

['Precipitation in the form of liquid water drops with diameters greater than 0.5 millimeters.', 'To fall from the clouds in drops of water.']

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))
```

O/P = Enter word: rain
[precipitation in the form of liquid water drops with diameters greater than 0.5 millimeters.', 'To fall from the clouds in drops of water.']

Taking into account Bad words:-

```
import json
```

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

```
def translate(w):
```

```
    if w in data:
```

```
        return data[w]
```

```
    else
```

```
        return "the word doesn't exist. Please double check it."
```

```
word = input("Enter word:")
```

```
print(translate(word))
```

O/P RAIN

The word doesn't exist. Please double check it.

Implementing Case Sensitivity:-

```
-> data["rain"]
```

```
"Rain".lower()
```

O/P 'rain'

```
import json
```

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

```
def translate(w):
```

```
    w = w.lower()
```

```
    if w in data:
```

```
        return data[w]
```

```
    else:
```

```
        return "the word doesn't exist. please double check it."
```

```
word = input("Enter word:")
```

```
print(translate(word))
```

IP: Rain rainn
the word doesn't exist. Please double check it.

Similarity Ratio Between two Words

```
-> import difflib  
-> from difflib import SequenceMatcher  
-> SequenceMatcher(None, 'rainn', 'rain').ratio()  
-> get_close_matches
```

Test matches out of a list of words:-

```
-> help(get_close_matches)  
-> get_close_matches('rainn', ['help', 'pyramid',  
                                'raion'])  
o/p ['rain']  
-> data = Keys()  
-> get_close_matches('rainn', data.keys(), n=5)  
['rain', 'train', 'rainy']
```

Recommending the Best Match:-

```
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:  
        return "Did you mean %s instead?" % get_close-
```

```
matches(w, data.keys())[0]
```

else:

return "the word doesn't exist. Please double check it."

word = input("Enter word:")

print(translate(word))

O/p rainn

Did you mean rain instead?

Confirmation from the user:-

```
yn = input("Did you mean %s instead? Enter  
Y if yes, or N if no: ")
```

close_matches(w, data
keys())[0]

if yn == "Y":

```
    return data[get_close_matches(w, data.keys()  
    if yn == "N":
```

O/p: rainn

Did you mean rain instead? Enter Y if yes, or

N if no: Y

-> return "the word doesn't exist. Please double check

else:

return "We didn't understand your entry."

else:

return "the word doesn't exist please
double check it."

```

1  # -*- coding: utf-8 -*-
2  import json
3
4  data = json.load(open("data.json"))
5
6  def translate(w):
7      w = w.lower()
8      if w in data:
9          return data[w]
10     else:
11         return "The word doesn't exist. Please double check it."
12
13 word = input("Enter word: ")
14
15 print(translate(word))
16
17
18 The word doesn't exist. Please double check it.
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

```

1  # -*- coding: utf-8 -*-
2  import json
3  from difflib import get_close_matches
4
5  data = json.load(open("data.json"))
6
7  def translate(w):
8      w = w.lower()
9      if w in data:
10         return data[w]
11     elif len(get_close_matches(w, data.keys())) > 0:
12         gm = input("Did you mean %s instead? Enter Y if yes, or N if no: ")
13         if gm == "Y":
14             return data[get_close_matches(w, data.keys())[0]]
15         elif gm == "N":
16             return "The word doesn't exist. Please double check it."
17         else:
18             return "We didn't understand your entry."
19     else:
20         return "The word doesn't exist. Please double check it."
21
22 word = input("Enter word: ")
23
24 print(translate(word))
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Version 1.1: Code

Lines 8 and 9 were added to make sure the program returns the definition of words that start with a capital letter (e.g. Delhi or Texas):

```

1  import json
2  from difflib import get_close_matches
3  data = json.load(open("data.json"))
4  def translate(w):
5      w = w.lower()
6      if w in data:
7          return data[w]
8      elif w.title() in data: #if user entered "Delhi" this will check
9          return data[w.title()]
10     elif len(get_close_matches(w, data.keys())) > 0:
11         gm = input("Did you mean %s instead? Enter Y if yes, or N if no: ")
12         if gm == "Y":
13             return data[get_close_matches(w, data.keys())[0]]
14         elif gm == "N":
15             return "The word doesn't exist. Please double check it."
16         else:
17             return "We didn't understand your entry."
18     else:
19         return "The word doesn't exist. Please double check it."
20
21 word = input("Enter word: ")
22 output = translate(word)
23 if type(output) == list:
24     for item in output:
25         print(item)
26 else:
27     print(output)
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

