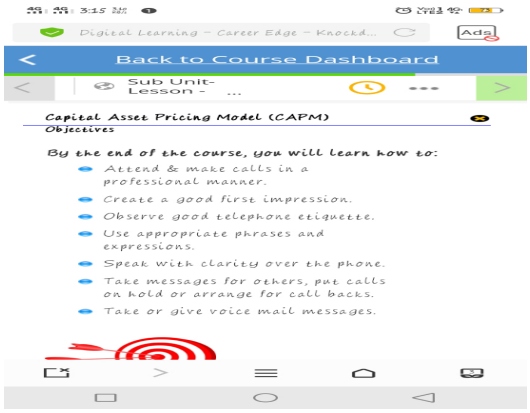
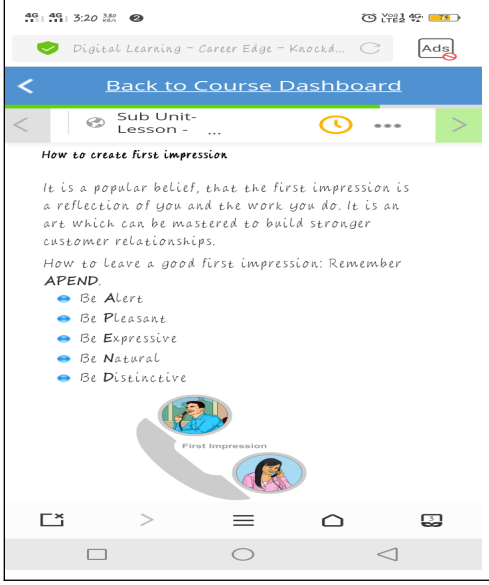
Day 4 report

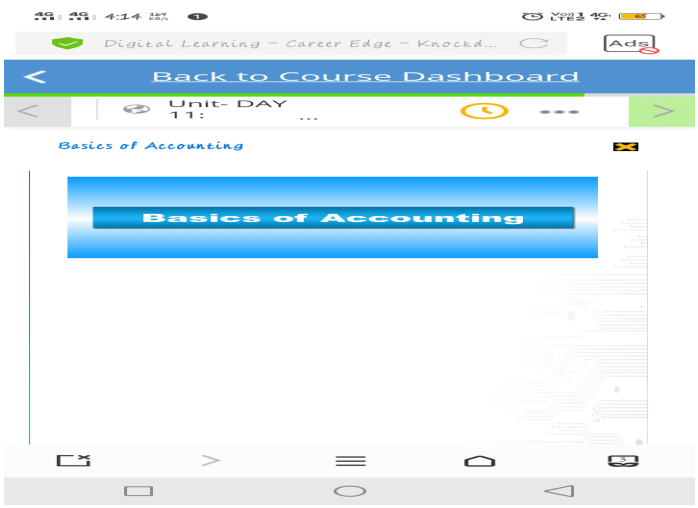
Date:	21-May-2020	Name:	Ankitha c c
Course:	TCS iON	USN:	4al16ec004
Topic:	1.Corporate telephone etiquette 2.Accounting fundamentals 3.Foundation skills in IT	Semester & Section:	8th sem & 'A' section
Github Repository:	ankitha-c-c		

MORNING SESSION DETAILS

Image of session







Report of the course:

1.Corporate telephone etiquette

However, when it comes to business introductions, sales pitches, or the handling of sensitive information, nothing beats a phone conversation. Even as technology advances, it is unlikely that business phone calls will go away anytime soon. So how do you ensure you're applying modern etiquette to this timeless form of communication? In order to ensure you are properly branding your company over the phone and providing exceptional customer service, we've assembled some do's and don'ts to guide your calls:

A. Business Phone Etiquette Do's:

Introduce yourself. Even in the age of caller ID, it can be jarring to pick up the phone and jump right into a conversation. Remind the person you're calling who you are and why you're reaching out. They will appreciate the courtesy even if they recognized your name and number when they picked up.

Speak clearly. Enunciation is so important when you don't have body language cues to pick up on. Be sure to speak carefully and slowly so that the person on the other end of the line can understand you.

Listen to requests. When you're busy multitasking, it can be hard to focus on the requirements of the person on the phone. Set other tasks aside as you actively listen to requests from the caller. Ask follow-up questions for clarification and to show that you're engaged in the conversation.

Make the kind of call you'd like to receive. The Golden Rule applies to just about everything in life, but this is especially true in the business world. Treat callers the way you like to be treated on the phone. That means making easy, friendly conversation that accomplishes goals for both parties.

Leave succinct voicemails. If you get someone's voicemail, leave a message that's short and sweet. Include your contact information and a quick indicator of why you're calling

B. Business Phone Etiquette Don'ts:

Don't answer the phone too casually in a business setting. A "Hello!" is fine but consider

including your name as you pick up a call. State the name of the business when you're answering the phone, too.
Don't leave people on hold for a long period of time. If you have to put a person on hold, ask them if it's okay to do so. At times, it may be better to take down the caller's number and ask to call them back later. There's no quicker way to make a person feel unimportant than by placing them on an extended hold!
Don't talk with your mouth full. Your mom probably warned you not to do this, and if you're making important calls over your lunch hour, you might be tempted to chat as you chew. Resist the urge! There's nothing worse than someone smacking their lips in your ear.
Don't use slang. Unless you know the person on the other end of the line socially, it's best to keep language professional. Swap out "hang on a sec" for "just a moment, please" and you're sure to impress the person on the line – or at least not offend them with your casual vibes.
Don't speak too loudly. Depending on the volume in the room you're in, you could be speaking far too loudly when making calls. Adjust your voice to a normal tone and don't be afraid to ask the person you're calling if they can hear you. Nobody likes being shouted at over the phone.

2.Accounting fundamentals

Accounting is the practice of recording and reporting on business transactions. The basics of accounting can be summarized within the following points:

System of record keeping. First, there must be a rational approach to record keeping. This means setting up accounts in which information is stored. Accounts fall into the following classifications:
Assets. These are items purchased or acquired, but not immediately consumed. Examples are accounts receivable and inventory.
Liabilities. These are obligations of the business, to be paid at a later date. Examples are accounts payable and loans payable.
Equity. This is assets minus liabilities, and represents the ownership interest of the owners of the business.
Revenue. This is the amount billed to customers in exchange for the delivery of goods or provision of services.
Expenses. This is the amount of assets consumed during the measurement period. Examples are rent expense and wages expense.
Transactions. The accountant is responsible for producing a number of business transactions, while others are forwarded to the accountant from other parts of the company. As part of these transactions, they are recorded within the accounts that we noted in the first point. Key transactions are:
Purchase materials and services. Requires the issuance of purchase orders and the payment of supplier invoices.
Sell goods and services to customers. Requires the creation of an invoice to be sent to each

customer, documenting the amount owed by the customer.
Receive payments from customers. Requires matching received cash to open invoices.
Pay employees. Requires the collection of time worked information from employees, which is then used to produce gross wage information, tax deductions, and other deductions, resulting in net pay to employees.
Reporting. Once all of the transactions related to an accounting period have been completed, the accountant aggregates the information stored in the accounts and reformats it into three documents that are collectively called the financial statements. These statements are:
Income statement. This document presents revenues and subtracts all expenses incurred to arrive at a net profit or loss for the reporting period. It measures the ability of a business to attract customers and operate in an efficient manner.
Balance sheet. This document presents the assets, liabilities, and equity of a business as of the end of the reporting period. It presents the financial position of an entity as of a point in time, and is closely reviewed to determine the ability of an organization to pay its bills.
Statement of cash flows. This document presents the sources and uses of cash during the reporting period. It is especially useful when the amount of net income appearing on the income statement varies from the net change in cash during the reporting period.

3.Foundation skills in IT

A. Objective

The objective of the course is to train students on basic skills in IT to help enhance employability and make students industry-ready. This interactive course is disseminated through experiential learning. The skills acquired through this course help the students prepare for employment, and as well orient them towards life-long learning. This interactive course is designed by experts from large member companies—Accenture, Cognizant, HCL, Infosys, Microsoft, TCS and Wipro (Mission 10X). PMI has provided the Project Management Module.

B. COURSE OUTLINE

The contents of this course are designed to support the course objectives. The following focus areas are included in this course:

Section A: Technology

Module 1: Technology Fundamentals
Module 2: RDBMS
Module 3: Software Development Life Cycle
Module 4: Networking
Section B: Business Dynamics

Module 5: Campus to Corporate

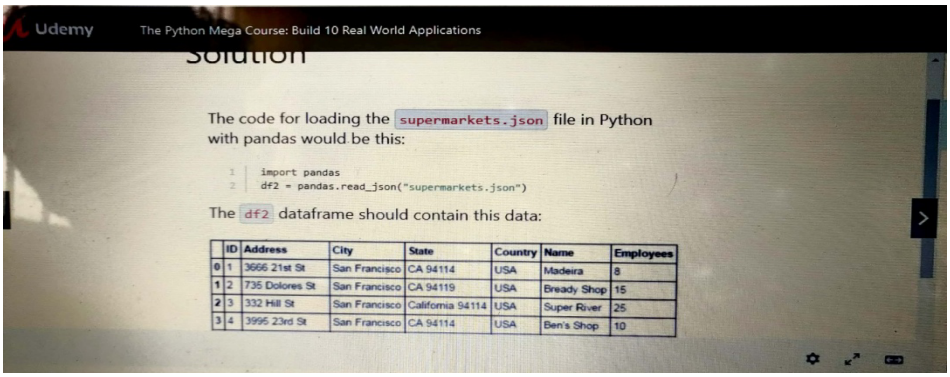
Module 6: Interpersonal Effectiveness
Section C: Principles of Project Management

Module 7: Setting up projects
Module 8: Project Planning
Module 9: Executing and managing a project

Date:	21-5-2020	Name:	Ankitha c c
Course:	Python programming	USN:	4AL16EC004
Topic:	Data analysis with panda	Semester & Section:	8th A
Github Repository:	ankitha-c-c		

AFTERNOON SESSION DETAILS

Image of session



Pandas is the most popular python library that is used for data analysis. It provides highly optimized performance with back-end source code is purely written in C or Python.
We can analyze panda by :

1)SERIES
2) DATAFRAME

Series:
Series is one dimensional(1-D) array defined in pandas that can be used to store any data type.

```
# Program to Create series with scalar values
Data =[1, 3, 4, 5, 6, 2, 9] # Numeric data

# Creating series with default index values
s = pd.Series(Data)

# predefined index values
Index =['a', 'b', 'c', 'd', 'e', 'f', 'g']

# Creating series with predefined index values
si = pd.Series(Data, Index)
```

Output:

```
In [4]: s
Out[4]:
0    1
1    3
2    4
3    5
4    6
5    2
6    9
dtype: int64
```

Panda is a Python library that provides extensive means for data analysis. Data scientists often work with data stored in table formats like .csv, .tsv, or .xlsx. Pandas makes it very convenient to load, process, and analyze such tabular data using SQL-like queries. In conjunction with Matplotlib and Seaborn, Pandas provides a wide range of opportunities for visual analysis of tabular data.

The main data structures in Pandas are implemented with **Series** and **DataFrame** classes. The former is a one-dimensional indexed array of some fixed data type. The latter is a two-dimensional data structure - a table - where each column contains data of the same type. You can see it as a dictionary of Series instances. DataFrames are great for representing real data: rows correspond to instances (examples, observations, etc.), and columns correspond to features of these instances.

In [1]:

```
import numpy as np
```

```
import pandas as pd
pd.set_option("display.precision", 2)
```

Understand the basic Pandas data structures

Pandas has two core data structures used to store data: The Series and the DataFrame.

Series

The series is a one-dimensional array-like structure designed to hold a single array (or 'column') of data and an associated array of data labels, called an *index*. We can create a series to experiment with by simply passing a list of data, let's use numbers in this example:

```
Copy contents
import pandas as pd

my_series = pd.Series([4.6, 2.1, -4.0, 3.0])
print(my_series)
```

The output should be:

```
Copy contents
0    4.6
1    2.1
2   -4.0
3    3.0
dtype: float64
```

Note that printing out our Series object prints out the values and the index numbers. If we just wanted the values, we can add to our script the following line:

```
Copy contents
print(my_series.values)
```

Which in addition will print:

```
Copy contents
array([ 4.6,  2.1, -4. ,  3. ])
```

For a lot of applications, a plain old *Series* is probably not a lot of use, but it is the core component of the Pandas workhorse, the DataFrame, so it's useful to know about.

DataFrames

The DataFrame represents tabular data, a bit like a spreadsheet. DataFrames are organised into colums (each of which is a Series), and each column can store a single data-type, such as floating point numbers, strings, boolean values etc. DataFrames can be indexed by either their row or column names. (They are similar in many ways to R's data.frame.)

We can create a DataFrame in Pandas from a Python dictionary, or by loading in a text file containing tabular data. First we are going to look at how to create one from a dictionary.

Setup

Let's create a pandas DataFrame with 5 columns and 1000 rows:

- a1 and a2 have random samples drawn from a normal (Gaussian) distribution,
- a3 has randomly distributed integers from a set of (0, 1, 2, 3, 4),
- y1 has numbers spaced evenly on a log scale from 0 to 1,
- y2 has randomly distributed integers from a set of (0, 1).

```
mu1, sigma1 = 0, 0.1
mu2, sigma2 = 0.2, 0.2
n = 1000df = pd.DataFrame(
   {
      "a1": pd.np.random.normal(mu1, sigma1, n),
      "a2": pd.np.random.normal(mu2, sigma2, n),
      "a3": pd.np.random.randint(0, 5, n),
      "y1": pd.np.logspace(0, 1, num=n),
      "y2": pd.np.random.randint(0, 2, n),
   }
)
```

Readers with Machine Learning background will recognize the notation where a1, a2 and a3 represent attributes and y1 and y2 represent target variables. In short, Machine Learning algorithms try to find patterns in the attributes and use them to predict the unseen target variable — but this is not the main focus of this blog post. The reason that we have two target variables (y1 and y2) in the DataFrame (one binary and one continuous) is to make examples easier to follow.

We reset the index, which adds the index column to the DataFrame to enumerates the rows.

```
df.reset_index(inplace=True)
```

	index	a1	a2	a3	y1	y2
0	0	0.049671	0.479871	2	1.000000	1
1	1	-0.013826	0.384927	2	1.002308	0
2	2	0.064769	0.211926	2	1.004620	0
3	3	0.152303	0.070613	3	1.006939	0
4	4	-0.023415	0.339645	4	1.009262	0