**DAILY ASSESSMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **21/05/2020** | **Name:** | **Dhavala** |
| **Course:** | **TCSion** | **USN:** | **4AL17EC027** |
| **Topic:** | * **Learn Corporate Telephone Etiquette** * **Understand Accounting Fundamentals** * **Gain Foundational Skills in IT** | **Semester & Section:** | **6TH SEM & A Section** |
| **Github Repository:** | **Dhavala27** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session**  **Learn Corporate Telephone Etiquette test** |
| **Report**  **Learn Corporate Telephone Etiquette**  In this module I learnt how to attend and make calls in a professional manner. Create a good first impression, observe good telephone etiquette, use appropriate phrases and expressions, speak with clarity over the phone, take message for others, put calls on hold or arrange for call backs.  **How to create first impression**  The first impression is reflection of you and the work you do. It is an art which can be mastered to build stronger customer relationships.   * Be Alert * Be Pleasant * Be Expressive * Be Natural * Be Distinctive   Do’s of telephone etiquette   * Identify yourself to the caller at the beginning of the call * Answer the call within 2 rings, with a smile. * Help the caller by providing correct information or transferring the call to the correct person and department * Be courteous and respectful to the caller * Use considerate phrases * Be as helpful as you can. * Ask the purpose of the call * Acknowledge the caller’s queries * Transfer the call if required   Don’ts of telephone etiquette   * Don’t bluff * Don’t speak negatively * Don’t sound weary * Don’t be impatient and rude * Don’t leave the caller on hold for long * Don’t speak to someone else when you answer the call * Don’t put the call on loud speaker mode * Don’t use slang * Don’t forget to end the call properly   Taking messages   * Take note on paper * Keep your stationery ready * Jot down important information * Write neatly * Verify the details of the contact person * Repeat the noted message for confirmation * Note the urgency of the message * Put your initials at the bottom of the message   **Voice mail etiquette**  Voicemails are digital recording of incoming and outgoing voice messages. The voicemail system allows the caller to convey a message even in the absence of the called person.  Features:   * Message can be taken from multiple callers * Message from the called person is conveyed to each person who calls * Message can be stored for a long time * Message can be stored externally locally or network or cloud storage   Essential guideline for telephone etiquette   * Answer the phone promptly * Identify your organization and then yourself * Show your genuine interest in the caller * Address their problems positively * Speak directly into the receiver * Listen patiently   Telephone courtesies   * Speak with enthusiasm * Use the caller’s name if it is known person * Be soft and polite * Avoid chewing gum while talking * Avoid putting the caller on hold unnecessarily * End the conversation with a positive note * Hang up on the caller gently   **Understand Accounting Fundamentals**  Accounting is a system which collects and processes financial information of a business.  Accounting is called as language of business. The need of business for recording transaction in systematic manner has given rise to book keeping. Only transactions related to business expressible in money terms are recorded.  The process of accounting is below.  Accounting assumption   * Accounting entity assumption * Money measurement assumption * Accounting period principle * Going concern assumption   Accounting Principles and concepts   * Dual aspect principle * Revenue Realization concept * Historical cost concept * Matching concept * Verifiable and objective evidence concept   Modifying principles   * Cost benefit principle * Materiality principle * Consistency principle * Prudence principle   Adjusting entries   * Outstanding expenses * Prepaid expenses * Accrued incomes * Incomes received in advance * Interest on loan * Forgiveness of loan * Accounts receivables/payables   Accounting standards  International accounting standards committee (IASC)   * Setup in 1973 * To formulate the accounting standards * It minimizes difference in accounting   **Gain Foundational Skills in IT**  What do recruiters expect   * Spend time on your final year project * Communicate your technical strengths upfront * Accept what you don’t know * Give logical paths to get to the solution * Communicate effectively |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date:** | **21/05/2020** | | **Name:** | **Dhavala** | |
| **Course:** | **Python** | | **USN:** | **4AL17EC027** | |
| **Topic:** | * **Project Exercise with Python and MySQL: Interactive English Dictionary** * **Data Analysis with Pandas** | | **Semester & Section:** | **6TH SEM & A Section** | |
| **Github Repository:** | **Dhavala27** | |  |  | |
|  | |
|  | |
|  | |
| **AFTERNOON SESSION DETAILS** | | | | |
| **Image of session** | | | | |

|  |
| --- |
| **Report**  **Project Exercise with Python and MySQL: Interactive English Dictionary**  SQL statement in Python code:  query = cursor.execute("SELECT \* FROM Dictionary WHERE Expression = 'rain'")  That statement retrieved all the rows of the Dictionary table where the value of the column Expression was rain. The string inside cursor.execute() is SQL code that Python sends to the database. That kind of language is understood by the database.  Here are some more examples of SQL queries that you can try out from within your Python script just like we did previously:  Get all rows where the value of the column Expression starts with r:  "SELECT \* FROM Dictionary WHERE Expression  LIKE 'r%'"  Get all rows where the value of the column Expression starts with rain:  "SELECT \* FROM Dictionary WHERE Expression  LIKE 'rain%'"  All rows where the length of the value of the column Expression is less than four characters:  "SELECT \* FROM Dictionary WHERE length(Expression) < 4"  All rows where the length of the value of the column Expression is four characters:  "SELECT \* FROM Dictionary WHERE length(Expression) = 4"  All rows where the length of the value of the column Expression is greater than 1 but less than 4 characters:  "SELECT \* FROM Dictionary WHERE length(Expression) > 1 AND length(Expression) < 4"  All rows of column Definition where the value of the column Expression starts with r:  "SELECT Definition FROM Dictionary WHERE Expression LIKE 'r%'"  **Data Analysis with Pandas**  Pandas is a library of providing data structures and data analysis tools within Python or if the  word tools confuse you, then you can say pandas is a library providing data structures and data analysis code.  So basically, pandas allow to load data from different sources into python and then use python code to analyse those data and produce results which can be in the form of tables, text, and also visualization.  **Installing Pandas**  Make sure you have pandas installed. You can install it with pip:  Pip install pandas  Or  Pip3 install pandas  Also, in the next lecture, we will use an enhanced Python interactive shell called IPython.  IPython is just like the normal shell you get when you run python, but IPython provides better printing for large text. This ability makes IPython suitable for data analysis because the program prints data in a well-structured format. You can install IPython with pip:  Pip install IPython  Or  Pip3 install IPython  Ex:  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):  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 w.title() in data: #if user entered "texas" this will check for "Texas" as well.  return data[w.title()]  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)  You can read the comments I included in the code: I have added another conditional in lines 8 and 9. The w.title() method will convert the first letter to uppercase and the rest to lowercase. If the program didn't find anything for "texas" in the first conditional in lines 6 and 7, then this conditional will try to search for "Texas". Even if the user entered "TEXAS" this conditional will convert it to "Texas".  Another ex:   added lines 10 and 11 to make sure the program returns the definition of acronyms (e.g., USA or NATO.)  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 w.title() in data:  return data[w.title()]  elif w.upper() in data: #in case user enters words like USA or NATO  return data[w.upper()]  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) |