**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **20/5/2020** | **Name:** | **Divyashree lv** |
| **Course:** | **TCS ION** | **USN:** | **4AL17EC030** |
| **Topic:** | **1. Gain Guidance from Career Gurus**  **2. Write a Winning Resume and Cover Letter** | **Semester & Section:** | **6th-‘A’** |
| **Github Repository:** | **div** | **E-mail:** | **divyagowdalv@gmail.com** |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session** |
| **Report – Report can be typed or hand written for up to two pages.**  **Write a Winning resume and Cover letter:**  Objectives:   * Explain the importance of resume. * Describe the structure of a resume. * State the Do’s and Don’ts of resume writing. * Write your own resume. * Discuss the content of a cover letter. * State the Do’s and Don’ts of the cover letter. * Write your own cover letter.   Importance of resume:   * Your resume has the power to get you an interview for your dream job. * The very same resume can be the reason for your dream job just remaining a dream. * Organizations use the resume to sort list potential candidates. * Your resume is the virtual YOU. It should speak of your passion and the career you want to follow.   Types of Resume:   * Chronological * Functional * Combinational   Importance of Cover letter:   * A cover letter is a read before your resume is read. * Each cover letter should be tailor made to the job you are seeking. * The cover letter tells the employer the role that you are interested in. * It tells the employer how qualified you are for the role. * A cover letter expresses points that your resume might not cover.     **Stay ahead in Group Discussion:**  **Objectives:**   * Explain why and how a group discussion is conducted. * Actively participate in GD. * Use some effective phrases in a GD.   What is a GD?   * It is a positively exchange of views on a particular topic. * It is a time bound and open to all present in the group. * It is not a competition or an argument where any one person wins. * It is often used for mass grading process.   Why is a GD conducted?  Group Discussions are conducted to test your,   * Communication skills * Ability to work in a team * Ability to express your point of view * Listening skills * Ability to handle criticism * Knowledge of the subject * Overall personality   Do’s:   * Dress formally. * Maintain eye contact with all the group members. * Have a neutral tone of voice. * Try to initiate the GD. * Express your point of view at the earliest. * Listen to all the points made * Be open minded about the topics.   Don’ts:   * Don’t let a small group to dominate the discussion. * Don’t let discussion drift away from the given topic. * Don’t interrupt the mid-sentence. * Don’t take any topic personally. * Don’t force others to think the way you do. * Don’t agree with all points of view. * Don’t get biased. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **20/5/2020** | **Name:** | **Divyashree LV** |
| **Course:** | **UDEMY-The Python Mega Course: Build 10 real world applications** | **USN:** | **4AL17EC030** |
| **Topic:** | **i. List Comprehensions**  **ii. More of Functions**  **iii. File Processing**  **iv. Imported modules** | **Semester & Section:** | **6th-‘A’** |
| **AFTERNOON SESSION DETAILS** | | | |
| **Image of session**    **Report – Report can be typed or hand written for up to two pages.**   1. **List of Comprehensions:**  * A list comprehension is an expression that creates a list by iterating over another container. * A **basic**list comprehension:   1. [i\*2 for i in [1, 5, 10]]   Output: [2, 10, 20]   * List comprehension with **if** condition:   1. [i\*2 for i in [1, -2, 10] if i>0]   Output: [2, 20]   * List comprehension with an **if** **and** **else** condition:   1. [i\*2 if i>0 else 0 for i in [1, -2, 10]]   Output: [2, 0, 20]   1. **More on Functions:**  * Functions can have more than one **parameter**:  1. def volume(a, b, c): 2. return a \* b \* c  * Functions can have **default** parameters (e.g. coefficient):  1. def converter(feet, coefficient = 3.2808): 2. meters = feet / coefficient 3. return meters 5. print(converter(10))   Output: 3.0480370641306997   * Arguments can be passed as **non-keyword** (positional) arguments (e.g. a) or **keyword** arguments (e.g. b=2 and c=10):  1. def volume(a, b, c): 2. return a \* b \* c 4. print(volume(1, b=2, c=10))  * An **\*args**parameter allows the  function to be called with an arbitrary number of non-keyword arguments:  1. def find\_max(\*args): 2. return max(args) 3. print(find\_max(3, 99, 1001, 2, 8))   Output: 1001   * An **\*\*kwargs** parameter allows the function to be called with an arbitrary number of keyword arguments:  1. def find\_winner(\*\*kwargs): 2. return max(kwargs, key = kwargs.get) 4. print(find\_winner(Andy = 17, Marry = 19, Sim = 45, Kae = 34))   Output: Sim   1. **File processing:**  * You can **read** an existing file with Python:  1. with open("file.txt") as file: 2. content = file.read()  * You can **create** a new file with Python and **write** some text on it: * with open("file.txt", "w") as file: * content = file.write("Sample text") * You can **append** text to an existing file without overwriting it:  1. with open("file.txt", "a") as file: 2. content = file.write("More sample text")  * You can both **append and read** a file with:  1. with open("file.txt", "a+") as file: 2. content = file.write("Even more sample text") 3. file.seek(0) 4. content = file.read() 5. **4. Imported Modules:**  * **Builtin objects** are all objects that are written inside the Python interpreter in C language. * **Builtin modules** contain builtins objects. * Some builtin objects are not immediately available in the global namespace. They are parts of a builtin module. To use those objects the module needs to be **imported** first. E.g.:   1. import time   2. time.sleep(5) * **A list of all builtin modules** can be printed out with:   1. import sys   2. sys.builtin\_module\_names * **Standard libraries** is a jargon that includes both builtin modules written in C and also modules written in Python. * **Standard libraries** written in Python reside in the Python installation directory as *.py* files. You can find their directory path with sys.prefix. * **Packages** are a collection of *.py* modules. * **Third-party libraries** are packages or modules written by third-party persons (not the Python core development team). * Third-party libraries can be **installed** from the terminal/command line:   Windows:  pip install pandas or use python -m pip install pandas if that doesn't work.   * Mac and Linux:   pip3 install pandas or use python3 -m pip install pandas if that doesn't work. | | | |