**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **19/05/2020** | **Name:** | **Krishna Swetha** |
| **Course:** | **TCS-ION** | **USN:** | **4AL16EC032** |
| **Topic:** | **Gain Guidance from Career Gurus Write a Winning Resume and Cover Letter**  **Stay Ahead in Group Discussions** | **Semester & Section:** | **6th, B section** |
| **Github Repository:** | **Krishna-Swetha** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
|  |
| **Report – Report can be typed or hand written for up to two pages.** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:** | **19/05/2020** | **Name:** | **Krishna Swetha** | |
| **Course:** | **Python** | **USN:** | **4AL16EC032** | |
| **Topic:** | **List Comprehensions**  **More on Functions**  **File Processing**  **Imported Modules** | **Semester & Section:** | **6th, B section** | |
| **AFTERNOON SESSION DETAILS** | | | |
|  | | | |
| **Report – Report can be typed or hand written for up to two pages.**  **Summary: List Comprehensions**  **In this section I learned that:**  **A list comprehension is an expression that creates a list by iterating over another container.**  **A basic list comprehension:**  **[i\*2 for i in [1, 5, 10]]**  **Output: [2, 10, 20]**  **List comprehension with if condition:**  **[i\*2 for i in [1, -2, 10] if i>0]**  **Output: [2, 20]**  **List comprehension with an if and else condition:**  **[i\*2 if i>0 else 0 for i in [1, -2, 10]]**  **Output: [2, 0, 20]**  **Python was the sixth most popular programming language in 2010 on StackOverflow**  **It continually went up in the ranks to being the most popular among all in 2018**    **Summary: More on Functions**  **In this section I learned that:**  **Functions can have more than one parameter:**  **def volume(a, b, c):**  **return a \* b \* c**  **Functions can have default parameters (e.g. coefficient):**  **def converter(feet, coefficient = 3.2808):**  **meters = feet / coefficient**  **return meters**    **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):**  **def volume(a, b, c):**  **return a \* b \* c**    **print(volume(1, b=2, c=10))**  **An \*args parameter allows the function to be called with an arbitrary number of non-keyword arguments:**  **def find\_max(\*args):**  **return max(args)**  **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:**  **def find\_winner(\*\*kwargs):**  **return max(kwargs, key = kwargs.get)**    **print(find\_winner(Andy = 17, Marry = 19, Sim = 45, Kae = 34))**  **Output: Sim**  **Here's a summary of function elements:**    **Summary: File Processing**  **In this section I learned that:**  **You can read an existing file with Python:**  **with open("file.txt") as file:**  **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:**  **with open("file.txt", "a") as file:**  **content = file.write("More sample text")**  **You can both append and read a file with:**  **with open("file.txt", "a+") as file:**  **content = file.write("Even more sample text")**  **file.seek(0)**  **content = file.read()**  **Summary: Imported Modules**  **In this section I learned that:**  **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.:**  **import time**  **time.sleep(5)**  **A list of all builtin modules can be printed out with:**  **import sys**  **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.** | | | |