**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **18/05/2020** | **Name:** | **Dhavala** |
| **Course:** | **TCSion** | **USN:** | **4AL17EC027** |
| **Topic:** | **Communicate to Impress**  **Deliver Presentations with Impact**  **Develop Soft Skills for the Workplace** | **Semester & Section:** | **6TH SEM & A Section** |
| **Github Repository:** |  |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session**  **1.Communicate to Impress Test**    **2.Deliver Presentations with Impact Test**    **3.Develop Soft Skills for the Workplace Test** |
| **Report**  **1.Communicate to Impress**  In this module I have learnt about the importance of communication, process of communication, barriers of communication, distinguish between verbal and non-verbal communication, use of communication effectively.  **What is communication?**  Communication is an act of sending information from one person to another. The definition may seem simple but the actual process in complex.  **Types are:**  Verbal, Nonverbal, Visual, Written.  Process on communication involves sender, encoder, channel, decoder, receiver, feedback.  **Barriers**  Barriers to communication causes disturbance in communication.  **Types of barriers are:**  Language, Perceptual, Physical, Cultural, Gender, Psychological.  **Non-verbal communication**  Facial expressions, Paralanguage, Gestures, Posture, Eye-contact, Appearance.  **Facial expressions:** Facial expressions play an important role in conveying a message. There has to be synchronization between your expressions and what you say, else people will subconsciously respond to the facial expression.  **Paralanguage:** It includes Tone, Clarity, Pace, Volume of the person who is communication.  **Gestures:** Gestures are physical actions which communicate with the onlooker. Gestures are made by moving the parts of your body, arms and hands.  **Posture:** Posture is how we sit or stand. Based on our posture a person looking at us can determine how we feel.  **Eye contact:** Maintaining eye contact a conversation signals that we are interested in the topic and are paying attention.  **Appearance:** The clothes you wear play an important role in nonverbal communication  **Verbal communication.**  Verbal communication plays an important role in effective communication and makes sure that the sender’s message reaches the receiver without too much barriers.  **Effective Communication**  Effective communication arises when verbal and non-verbal communication work together.  **2.Deliver Presentations with Impact**  In this module I learnt how to design effective ppt’s, to make effective presentation, do and don’ts in presentation.    **Designing effective presentation**  We should be knowing 5W’s while creating presentation.  **Those are:** What, Who, Why, Where, When.  **Number of slides**  We should make few slides; a greater number of slides make complex and the attention of participants will be lost.  **Fewer words**  We use a single word or sentence and then elaborate as you present and look at the example to understand better.  **Simple language**  Jargon and slang should be avoided and language used should be simple and understood by all, abbreviated words should be used only after using the full form for the first time.  There should be few images per slides.  There should be only one thought per slide.  Final slide should say thank you and ask for any doubts.  Fonts should be readable.  Avoid uppercase and use attractive slides.  Sounds and animation should be put only if it is necessary.  Background should be simple.  **Make an effective presentation**  We need to follow up few preparations before the presentation  Make notes of content.  Rehearse the presentation.  Gather all the materials  Feedback.  **3.Develop Soft Skills for the Workplace**  In this module I studied the importance of soft skills and distinguish between hard and soft skills.  **What are soft skills?**  Soft skills are people’s skills that are difficult to quantify and measure. These skills help us to maintain healthy relationships.  **It includes**  Communication skills, Time Management, Negotiation skills, Critical thinking, Self-confidence, Business etiquette, Goal setting, Team work.  **Need of soft skills**  Most of people who are good at their trade are ambitious and want to rise through the ranks. But they are unhappy at work or their supervisor thinks they are unfit or cannot handle the job. During interviews many people technically prove themselves, are passionate about the job are not selected.  Soft skill helps us in expressing our thoughts, work in a team, interview handling, ability to empathize, self-awareness, learning and sharing, interpersonal skills, builds confidence, good work ethic.  Hard skills and soft skills combine and makes a carrier growth. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date:** | **18/05/2020** | | **Name:** | **Dhavala** | |
| **Course:** | **TCSion** | | **USN:** | **4AL17EC027** | |
| **Topic:** | * **Introduction** * **The Basics: Small Program** * **The Basics: Data Types** * **The Basics: Operations with Data Types** * **The Basics: Functions and Conditionals** * **The Basics: Processing User Input** * **The Basics: Loops** | | **Semester & Section:** | **6TH SEM & A Section** | |
| **Github Repository:** |  | |  |  | |
|  | |
|  | |
|  | |
| **AFTERNOON SESSION DETAILS** | | | | |
| **Image of session** | | | | |
| **Report**  **1.Course introduction**  Python itself is a language, or to put it differently, a set of rules defined to enable the communication between us and computers. So, when we say print (1 + 2) that code is a rule the Python developers have agreed upon with computers to ask computers to print out the sum of 1 and 2.  **Preview of the 10 Apps**  There will be 10 interesting Python applications that we will build together in this course.  **Command-line-based Interactive English Dictionary**  **Web Maps on the Browser**  **Python Blocker of Distracting Websites**  **Portfolio Website**  **Desktop Bookstore App**  **Webcam Motion Detector and Plotter**  **Web scraping Program**  **Data Visualizer on the Browser**  **Web App with Database Backend and Email Sending Feature**  **Web App with Download-Upload Feature**  Then we studied about the tools required and the procedure how to install Python in Linux, Windows and Mac.  **2.The Basics: Small Program**   * Python 3 and the Visual Studio Code IDE is used in the videos, but you can use any IDE. * The Python interactive shell (shown with >>>) is a quick way to execute Python code to see how it works. * Python programs are written in *.py* files. * You can make a program that shows the current date and time using these lines of code: * import datetime * x = datetime.datetime.now() * print(x)   **3.The Basics: Data Types**  You can create a list of numbers automatically using a range. For example:  list (range(1,10))  That will output:  [1 2 3 4 5 6 7 8 9]  As you can see we just needed to specify the list boundaries inside range(). So, we specified 1and 10. Note that 10 is not included in the list. The generated list always runs up to one number before the upper number. In our example it goes up to 9 since the upper number is 10.  You can also specify a step as a third argument:  List(range(1,10,2))  That will output:  [1, 3, 5, 7, 9]  So, the count happens every two items starting from 1 and ending at 9.   * **Integers** are for representing whole numbers: * rank = 10 * eggs = 12 * people = 3 * **Floats** represent continuous values: * temperature = 10.2 * rainfall = 5.98 * elevation = 1031.88 * **Strings** represent any text: * message = "Welcome to our online shop! * "name = "John" * serial = "R001991981SW" * **Lists** represent arrays of values that may change during the course of the program: * members = ["Sim Soony", "Marry Roundknee", "Jack Corridor"] * pixel\_values = [252, 251, 251, 253, 250, 248, 247] * **Dictionaries** represent pairs of keys and values: * phone\_numbers = {"John Smith": "+37682929928", "Marry Simpons": "+423998200919"} * volcano\_elevations = {"Glacier Peak": 3213.9, "Rainer": 4392.1} * **Keys** of a dictionary can be extracted with: * phone\_numbers.keys() * **Values** of a dictionary can be extracted with: * phone\_numbers.values()   **Tuples** represent arrays of values that are not to be changed during the course of the program:  vowels = ('a', 'e', 'i', 'o', 'u')  one\_digits = (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)   * To find out what **attributes** a type has: * dir(str) * dir(list) * dir(dict)   **4.The Basics: Operations with Data Types**   * Lists, strings, and tuples have a **positive index** system: * ["Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun"] * 0 1 2 3 4 5 6 * **negative index** system: * ["Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun"] * -7 -6 -5 -4 -3 -2 -1 * In a list, the **2nd**, **3rd**, and **4th** items can be accessed with: * days = ["Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun"] * days[1:4] * Output: ['Tue', 'Wed', 'Thu']  1. **First three items of a list:** 2. days = ["Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun"] 3. days[:3] 4. Output:['Mon', 'Tue', 'Wed']  * A single in a **dictionary**can be accessed using its key: * phone\_numbers = {"John Smith":"+37682929928","Marry Simpons":"+423998200919"} * phone\_numbers["Marry Simpsons"] * Output: '+423998200919'   **5.The Basics: Functions and Conditionals**  Bonus Code: Using "and" and "or" in a Conditional   to check for one single condition:   1. x = 1 2. if x == 1: 3. print("Yes") 4. else: 5. print("No")   You can also check if two conditions are met at the same time using an and operator:   1. x = 1 2. y = 1 4. if x == 1 and y==1: 5. print("Yes") 6. else: 7. print("No")   That will return yes since x==1 and y==1 are both True.  You can also check if one of two conditions are met using an or operator:   1. x = 1 2. y = 1 4. if x == 1 or y==2: 5. print("Yes") 6. else: 7. print("No")   That will return yes since at least one of the conditions is True. In this case x==1 is True.   * Define a **function**: * def cube\_volume(a): * return a \* a \* a * Write a **conditional**block: * message = "hello there" * if "hello" in message: * print("hi") * else: * print("I don't understand") * Write a conditional block of **multiple conditions**: * message = "hello there" * if "hello" in message: * print("hi") * elif "hi" in message: * print("hi") * elif "hey" in message: * print("hi") * else: * print("I don't understand") * Use the and operator to check if **both conditions** are True at the same time:  1. x = 1 2. y = 1 4. if x == 1 and y==1: 5. print("Yes") 6. else: 7. print("No")   Output is yes since both x and y are 1.   * Use the or operator to check if **at least one condition** is True: * x = 1 * y = 2 * if x == 1 or y==2: * print("Yes") * else: * print("No")   Output is yes since x  is 1.  **5.The Basics: Processing User Input**  A Python program can get **user input** via the input function:   * The **input** **function** halts the execution of the program and gets text input from the user**:** * name = input("Enter your name: ") * The input function converts any **input to a string**, but you can convert it back to int or float: * experience\_months = input("Enter your experience in months: ") * experience\_years = int(experience\_months) / 12 * **format strings** with (works both on Python 2 and 3): * name = "Sim" * experience\_years = 1.5 * print("Hi %s, you have %s years of experience." % (name, experience\_years))   Output: Hi Sim, you have 1.5 years of experience.  **format strings** with (Python 3 only):   1. name = "Sim" 2. experience\_years = 1.5 3. print("Hi {}, you have {} years of experience".format(name, experience\_years))   Output: Hi Sim, you have 1.5 years of experience.  **7.The Basics: Loops**  For Loop Over a Function  using loops you can call any function multiple times, even your own functions. Let's suppose we defined this function:   1. def celsius\_to\_kelvin(cels): 2. return cels + 273.15   That is a function that gets a number as input, adds 273.15 to it and returns the result. A for loop allows us to execute that function over a list of numbers:   1. monday\_temperatures = [9.1, 8.8, -270.15] 3. for temperature in monday\_temperatures: 4. print(celsius\_to\_kelvin(temperature))   The output of that would be:  282.25  281.95  3.0  So, in the first iteration Celsius\_to\_kelvin(9.1) was executed, in the second Celsius\_to\_kelvin(8.8) and in the third Celsius\_to\_kelvin(-270.15)  That's just something to keep in mind.   * **For loops** are useful for executing a command over a large number of items. * You can create a **for loop** like so:  1. for letter in 'abc': 2. print(letter.upper())   Output:  A  B  C   * You can loop over **dictionary keys**:  1. phone\_numbers = {"John Smith":"+37682929928","Marry Simpons":"+423998200919"} 2. for value in phone\_numbers.keys(): 3. print(value)   Output:  John Smith  Marry Simpsos  Marry Simpsons   * You can loop over **dictionary values**:  1. phone\_numbers = {"John Smith":"+37682929928","Marry Simpons":"+423998200919"} 2. for value in phone\_numbers.values(): 3. print(value)   Output:  +37682929928  +423998200919   * You can loop over **dictionary items**:   1. phone\_numbers = {"John Smith":"+37682929928","Marry Simpons":"+423998200919"}   2. for key, value in phone\_numbers.items():   3. print(key, value)   Output:  (‘John Smith’.’+37682929928’)  (‘Marry Simpons’, ‘+423998200919’)   * **While loops** will run as long as a condition is true:   1. while datetime.datetime.now() < datetime.datetime(2090, 8, 20, 19, 30, 20):   2. print("It's not yet 19:30:20 of 2090.8.20")   The loop above will print out the string inside print() over and over again until the 20th of August, 2090. | | | | |