**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **29 may 2020** | **Name:** | **Veronica gudagur** |
| **Course:** | **python** | **USN:** | **4al16ec091** |
| **Topic:** | **udemy** | **Semester & Section:** | **8-B** |
| **Github Repository:** | **Veronica-g** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session** |
| Object-Oriented Programming (OOP) Object-oriented Programmingis a [programming paradigm](http://en.wikipedia.org/wiki/Programming_paradigm) which provides a means of structuring programs so that properties and behaviors are bundled into individual objects.  For instance, an object could represent a person with a name property, age, address, etc., with behaviors like walking, talking, breathing, and running. Or an email with properties like recipient list, subject, body, etc., and behaviors like adding attachments and sending.  Put another way, object-oriented programming is an approach for modeling concrete, real-world things like cars as well as relations between things like companies and employees, students and teachers, etc. OOP models real-world entities as software objects, which have some data associated with them and can perform certain functions. To Define a Class in Python Defining a class in Python:  classDog:  pass  You start with the class keyword to indicate that you are creating a class, then you add the name of the class **Class Attributes** While instance attributes are specific to each object, class attributes are the same for all instances—which in this case is all dogs.  classDog:  # Class Attribute  species='mammal'  # Initializer / Instance Attributes  def\_\_init\_\_(self,name,age):  self.name=name  self.age=age  example  class Dog:  # Class Attribute  species = 'mammal'  # Initializer / Instance Attributes  def \_\_init\_\_(self, name, age):  self.name = name  self.age = age  # Instantiate the Dog object  philo = Dog("Philo", 5)  mikey = Dog("Mikey", 6)  # Access the instance attributes  print("{} is {} and {} is {}.".format(  philo.name, philo.age, mikey.name, mikey.age))  # Is Philo a mammal?  if philo.species == "mammal":  print("{0} is a {1}!".format(philo.name, philo.species))  **Output**  Philo is 5 and Mikey is 6.  Philo is a mammal! Instance Methods Instance methods are defined inside a class and are used to get the contents of an instance. They can also be used to perform operations with the attributes of our objects. Like the \_\_init\_\_ method, the first argument is always self: |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |