A1)

object-oriented Programming (OOPs) is a programming paradigm that uses objects and classes in programming. It aims to implement real-world entities like inheritance, polymorphisms, encapsulation, etc. in the programming. The main concept of OOPs is to bind the data and the functions that work on that together as a single unit so that no other part of the code can access this data.

A2)

it searches a tree of linked objects, looking for the first appearance of attribute that it can find. When classes are involved, the preceding Python expression effectively translates to the following in natural language:Find the first occurrence of attribute by looking in object, then in all classes above it, from bottom to top and left to right.

A3)

**Class** is a detailed description, the definition, and the template of what an object will be. But it is not the object itself.

**Object** is an instance of a class. All data members and member functions of the class can be accessed with the help of objects.

A4)The first parameter in the class method is the class on which you are calling the method, not (necessarily) the class that defines the method.

A5)

All classes have a function called \_\_init\_\_(), which is always executed when the class is being initiated.We use the \_\_init\_\_() function to assign values to object properties, or other operations that are necessary to do when the object is being created.

A6)

class Pokemon:

# Class attribute

species = "Mouse"

def \_\_init\_\_(self, name, attack):

self.name = name

self.attack = attack

# One instance method

def description(self):

return f"{self.name} favorite attack is {self.attack}"

# A second instance method

def speak(self, sound):

return f"{self.name} says {sound}"

A7)

class Pokemon:

# Class attribute

species = "Mouse"

def \_\_init\_\_(self, name, attack):

self.name = name

self.attack = attack

A8)

A superclass is the class from which many subclasses can be created. The subclasses inherit the characteristics of a superclass. The superclass is also known as the parent class or base class.