1.What is Object-Oriented Programming, and how does it differ from procedural programming?

In object-oriented programming, the program is divided into small parts called objects. Objects contain data in the form of attributes and code in the form of methods.

In procedural programming, the program is divided into small parts called functions.

2.Explain the principles of OOP and how they are implemented in Python.

Describe the concepts of encapsulation, inheritance, and polymorphism in Python.

Object-Oriented Programming (OOP) is based on four fundamental principles: Encapsulation, Abstraction, Inheritance, and Polymorphism

Encapsulation is the principle of bundling data (attributes) and methods/functions that operate on the data into a single unit (i.e., an object). It hides the internal state of an object and restricts access to certain parts.

Inheritance allows a new class (derived or child class) to inherit properties and behaviors from an existing class (base or parent class). It facilitates code reuse and the creation of a hierarchy of classes.

Polymorphism refers to the ability of different objects or classes to provide a common interface or method signature, but with different implementations.

3. What is the purpose of the self keyword in Python class methods?

When we create an object from a class, the object is automatically passed as the first argument using the self parameter . This enables you to modify the object’s properties and execute tasks unique to that particular instance

5.What is the difference between class and instance variables in Python?

Class Variables:

Associated with the class itself, shared among all instances.

Defined outside of any instance method within the class.

Accessed using the class name (ClassName.variable\_name) or instances (self.variable\_name or instance\_name.variable\_name).

Modifications to class variables affect all instances simultaneously.

Often used for storing shared data across instances.

Instance Variables:

Belong to specific instances/objects of the class.

Defined using self within instance methods or the class constructor (\_\_init\_\_ method).

Accessed using self.variable\_name within instance methods or instance\_name.variable\_name.

Unique to each instance, changes to instance variables are isolated to that instance.

Hold data specific to individual instances, allowing each instance to have its own set of data.

6.Discuss the concept of abstract classes and how they are implemented in Python.

Abstract classes are used to define a common interface for a set of subclasses. They can define abstract methods, which are methods that have no implementation in the abstract class but must be implemented in the subclass

8. does Python support multiple inheritance, and what challenges can arise from it?

Python supports multiple inheritance, allowing a class to inherit attributes and methods from multiple parent classes. This means that a Python class can inherit from more than one base class. To implement multiple inheritance, a subclass can specify multiple parent classes in its class definition.

9.What is a decorator in Python, and how can it be used in the context of OOP?

A decorator takes in a function, adds some functionality and returns it

def make\_pretty(func):

def inner():

print ("I got decorated")

func()

return inner

def ordinary():

print ("I am ordinary ")

10.What do you understand by Descriptive Statistics? Explain by Example.

Descriptive statistics summarizes and describes data:

Central Tendency: Measures like mean, median, and mode show central values.

Variability/Spread: Range, variance, and standard deviation indicate data spread.

Distribution: Histograms, box plots, etc., illustrate data distribution.

Mean= 75+82+90+65+88+72+95+78+84+79/10 =80.8

Median= 79+82/2 =80.5

Mode: Determine the most frequent score.

Range = Max Score - Min Score

11. What do you understand by Inferential Statistics? Explain by Example

Inferential statistics is a branch of statistics that involves using data from a sample to make inferences or predictions about a larger population. It allows us to draw conclusions, make estimations, and test hypotheses about a population based on sample data.