The Questions of Test

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

Procedural programming can be defined as a programming model which is derived from structured programming, based upon the concept of calling procedure.It has ***top-down approach***.

Languages used in Procedural Programming: FORTRAN, ALGOL, COBOL,BASIC,C

Object-oriented programming can be defined as a programming model which is based upon the concept of objects. Objects contain data in the form of attributes and code in the form of methods. In object-oriented programming, computer programs are designed using the concept of objects that interact with the real world.it has ***bottom-up approach***. Object-oriented programming languages are various but the most popular ones are class-based, meaning that objects are instances of classes, which also determine their types.

Languages used in Object-Oriented Programming: *Java, C++, C#, Python*

2.Explain the principles of OOP and how they are implemented in Python.Describe the concepts of encapsulation, inheritance, and polymorphism in Python.

There are four major OOP principles in programming: encapsulation, abstraction, polymorphism, and inheritance.

Abstraction hides unnecessary code details from the users. data abstraction in python can be achived by creating abstract classes.

```
class employee(ABC):
    def emp_id(self,id,name,age):
        pass
class childemployee(employee):
    def emp_id(self,id):
        print("emp_id is 1")


emp1= childemployee()
emp1.emp_id(id)
```

Inheritance when a class derives from another class. the child class will inherits all the public and protected properties and methods from the parent class

```
class Parent():
    def first(self):
        print('first function')
class Child(Parent):
    def second(self):
        print('second function')
ob = Child()
ob.first()
ob.second()
```

Polymorphism -> poly means many and morphism means form

it refers to use the same functions name , but with different signatures, for multiple type

```
class bird():
  def __init__(self,color):
    self.color = color
  def features(self):
    print("can fly", self.color)
class ostrich():
  def __init__(self,color):
    self.color=color
  def features(self):
    print("can't fly", self.color)

def create_object(object):
    object.features()

create_object(bird("green"))
create_object(ostrich("black and white"))
```

Encapsulation wrapping data and the methods that work on data within one unit.

```
class students:
  def __init__(self,name,score):
    self.name=name
    self.score=score
  def demo(self):
    print("I am ",self.name)
    print("I score",self.score)
st1=students("nupriya",99)
st2=students("priya",100)
st1.demo()
st2.demo()
```

3.What is the purpose of the self keyword in Python class methods?
Self represents the instance of class.allows you to access variables, attributes, and methods of a defined class in Python.

4.How does method overriding work in Python, and why is it useful?
Method overriding is an ability of any object-oriented programming language that allows a subclass or child class to provide a specific implementation of a method that is already provided by one of its super-classes or parent classes.
5.What is the difference between class and instance variables in Python?
class variables are shared among all instances of a class, while instance variables are unique to each instance.
6.Discuss the concept of abstract classes and how they are implemented in Python.
A class containing one or more than one abstract method is called an abstract class.It allows you to create a set of methods that must be created within any child classes built from the abstract class.

```
from abc import ABC,abstractmethod
class employee(ABC):
    def emp_id(self,id,name,age):
        pass
class childemployee(employee):
    def emp_id(self,id):
        print("emp_id is 1")

emp1= childemployee()
emp1.emp_id(id)
```

7.Explain the importance of the super() function in Python inheritance.
The super() function in Python is a built-in function that allows you to call methods of a parent class from a child class. It is an essential tool for working with inheritance in Python, as it allows you to reuse code
8.How does Python support multiple inheritance, and what challenges can arise from it?
Python supports multiple inheritance, which means that a class can inherit from multiple parent classes.

```
class Mother:
    mothername = ""
    def mother(self):
        print(self.mothername)
class Father:
    fathername = ""
    def father(self):
```

```
        print(self.fathername)
class Son(Mother, Father):
    def parents(self):
        print("Father :", self.fathername)
        print("Mother :", self.mothername)
s1 = Son()
s1.fathername = "RAM"
s1.mothername = "SITA"
s1.parents()
```

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

it allows programmers to modify the behaviour of a function or class.
it allow us to wrap another function in order to extend the behaviour of the wrapped function, without permanently modifying it.

```
def deco(func):
    def inner():
        print("before function execution")
        func()
        print("after function execution")
    return inner
def func_used():
    print("inside the function")
func_used = deco(func_used)
func_used()
```

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

Descriptive statistics:-Summarizes key features like mean and median. It uses tables, graphs, and summary measures to organize, represent, and describe a collection of data.

measures of descriptive statistics:

Mean:

The mean is the average of a data set. It is calculated by adding up all the values in the data set and dividing by the number of values.

Median:

The median is the middle value in a data set that has been ordered from least to greatest. If there are two middle values, the median is the average of the two values.

Mode:

The mode is the most frequent value in a data set.

Range:

The range is the difference between the highest and lowest values in a data set.

Standard deviation:

The standard deviation is a measure of how spread out a data set is. It is calculated by taking the square root of the variance.

Variance:

The variance is a measure of how far each value in a data set is from the mean.

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

Predicts population trends from samples. It employs techniques to relate variables in a data set to one another. It involves hypothesis testing, a process of using statistical methods to determine whether the hypothesis about the population is likely true.

Hypothesis testing:-
A type of inferential statistics that involves setting up a null hypothesis and an alternative hypothesis. The goal is to compare populations or assess relationships between variables using samples.
Regression analysis:-
A statistical method that investigates the relationship between two or more variables. It aims to predict the value of a dependent variable based on one or more independent variables.
Confidence intervals:-
A range of values derived from the sample statistics that is likely to contain the value of an unknown population parameter.