1) Explain the importance of Functions?

Usages: Reusability , Clean Code , organised code , collaborative

3)What is the difference between print and return statements

Print statement only print the values. Whereas return statement is used return the value from the funtion and use it for other purpose;

What are \*args and \*\*kwargs ?

\*args is object which is used to take variable number of non dictionary parameter in function.

\*\*kwargs is used to take the variable number of input in dictionary format in function.

Explain the iterator function0

An object for list, tuple ,dictionary & sets. It is done by using iter method and next method, next method return the next element.

What is encapsulation in OOP?

Encapsulation is a Python technique for combining data and functions into a

single object. A class, for instance, contains all the data (methods and variables). Encapsulation refers to the

broad concealment of an object's internal representation from areas outside of its specification

Explain the use of access modifiers in Python classes?

: A class's data members and methods can be made private or protected in

order to achieve encapsulation. Direct access modifiers like public, private, and protected don't exist in Python,

though. Single and double underscores can be used to accomplish this.

Public Member : from outside of class, anywhere accessible.

Private Member : Within the class, accessible.

Protected Member : Within the class and its subclasses, accessible.

Public Member : Both inside and outside of a class, public data members are accessible. By default, the

class's member variables are all public

What is inheritance in OOP%?

Inheritence play a significant role in oop language. Inheritence in python refer to the process of a child receiving the parent class’s properties.

The reuse of code is inheritance’s main goal. Instaed of starting fron wratch when developing a new class. We can use the existing class instead of re\_creating it form scractch.

Define polymorphism in OOP?

Polymorphism in Object-Oriented Programming (OOP) refers to the ability of different objects to respond to the same function or method call in different ways. It allows objects of different classes to be treated as objects of a common super class, with each object responding in a way that is appropriate to its specific class.

Explain method overriding in Python?

Method overriding in Python is a feature that allows a subclass to provide a specific implementation of a method that is already defined in its superclass. When a method in a subclass has the same name, parameters, and return type as a method in its superclass, the method in the subclass overrides the one in the superclass.