Arguments

In Python, arguments are the actual values or data you pass into a function when you call it. They provide the necessary information for the function to perform its task.

Lambda function

A lambda function in python is concise way to define small, anonymous functions. Unlike regular functions defined using the def keyword, lambda functions are defined using the lambda keyword and are typically used for short-term tasks.

Syntax:

argument :- The input to the function

expression :- A single operation or calculation that the function performs

What is map() function

Imagine you have a list of items, like numbers, and you want to perform the same operation on each item (e.g., doubling each number) instead of writing a loop, you can use the map() function to apply an operation to all items in the list in one go.

In simple words we can say that map() function works on single - the single iterable not on more.

Filter() function

The filter function helps you select certain items from a list (or any iterable) based on a condition you specify. It is like using a sieve to keep only the items you want and discard the rest.

Key points

Purpose :- filter() helps you select item from a list based on a condition.

Return :- It returns a iterator, so you can convert it to a list or loop through it.

Condition :- The function you ~~need~~ pass should return True for items you want to keep and False for items to exclude.

Q1. What is class, objects, methods and attributes

Ans. A class is a collection of objects. A classes contains the blueprint for creating an object. A class defines a set of attributes and methods that created object.

Objects :- An object is an instance of class. It represent a specific implementation of the class and holds its own data.

Methods :- A method is a function that belongs to an object created from a class. It defines the behaviour or action an object can perform and can access or modify the object data.

A method is function defined inside a class and it designed to operate on instance of a class.

Attributes :- An attribute is a characteristic or quality that describes an object, person, or concept. It helps to define identify something by highlighting its specific features.

Q2) What is difference between class and objects.

Ans	class	objects
(i)	A blueprint or template for creating objects.	A specific instance created from class.
(ii)	Does not occupy memory until instantiated	Occupies memory when created
(iii)	Defines structure and behaviour	Holds unique data defined by attributes.
(iv)	Define structure and behaviour	Holds
(iv)	Defined using the class keyword	created by instantiating the class.
(v)	class Dog:	my_dog = Dog("Buddy", 3)

Q3) What does self refer to in a class method in python

Ans. Explain In python a self is a special variable used in a class method to refer to the current instance of the class. It allows each object created from the class to maintain its own attributes and methods.

Q4) Encapsulation and Inheritance with example.

Ans. Encapsulation is like putting your valuables in a secure box. You can access them through a controlled opening (methods), but you can't directly tamper with the contents.

Encapsulation involves bundling data (attributes) and methods (functions) that operate and that data into a single unit known as a class.

Why we use Encapsulation.

(i) Data protection:- Prevents unauthorized access or modification.

(ii) Controlled Access:- Allows changes only through designated methods.

(iii) Cleaner code:- Keeps internal details hidden, exposing only necessary parts

Inheritance:-

Inheritance is a fundamental concept in object-oriented programming that allows a class (called a child or subclass) to inherit attributes and methods from another class (called a parent or base class). It promotes code reuse and establishes a hierarchical relationship between classes

Why use inheritance:-

- Code Reusability:- Avoids redundant code by allowing new classes to reuse existing functionality.
- Modularity:- Organizes code into logical hierarchies, making it easier to maintain and extend
- Extensibility:- Facilities the addition of new features without modifying existing code.

Q5) what are instance attributes and how are they different from class attributes.

Ans Instance attributes are like personal details of an object. Each object created from a class can have its own unique set of instance attributes. These attributes are defined inside the class using the self keyword, ensuring that each object has its own copy.

for example

Instance attributes are like personal belongings of an object. Just as each person has their own unique set of belongings, each object has its own set of instance attributes that define its state.

Class Attributes

Class attributes are variables that belong to the class itself and are shared among all instances of the class.

Instance Attributes

(i) Defined inside __init__ using self.

(ii) Specific to each instance

(iii) Accessed via self.attribute_name

(iv) Changes affect only the specific instance

(v) Each instance has its own copy

Class Attributes

Defined directly within the class body.

Shared among all instances of the class

Accessed via .ClassName.attribute_name or self.attribute_name.

Changes affect all instances of the class.

Only one copy shared among all instances.