Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

AY: 2024-25

Class:	**SE**	**Semester:**	**IV**
Course Code:	**CSL405**	**Course Name:**	**Skills Based Python Programming Lab**

Name of Student:	**Shravani Sandeep Raut**
Roll No. :	**48**
Experiment No.:	**2**
Title of the Experiment:	**To implement functions and object oriented concepts in python.**
Date of Performance:	**28/01/2025**
Date of Submission:	**04/02/2025**

Evaluation

Performance Indicator	Max. Marks	Marks Obtained
Performance	5	
Understanding	5	
Journal work and timely submission	10	
Total	20	

Performance Indicator	Exceed Expectations (EE)	Meet Expectations (ME)	Below Expectations (BE)
Performance	4-5	2-3	1
Understanding	4-5	2-3	1
Journal work and timely submission	8-10	5-8	1-4

Checked by

Name of Faculty : Mr. Raunak Joshi

Signature :

Date:

Aim: To implement functions and object oriented concepts in python.

Theory:

Python allows us to divide a large program into the basic building blocks known as a function. The function contains the set of programming statements enclosed in a block. A function can be called multiple times to provide reusability and modularity to the Python program.

Creating a Function:
Python provides the def keyword to define the function. The syntax of the define function is given below.

Syntax:

1. def my_function(parameters):

2. function_block

3. return expression

Function Calling

In Python, after the function is created, we can call it from another function. A function must be defined before the function call; otherwise, the Python interpreter gives an error. To call the function, use the function name followed by the parentheses.

Recursive function

A function that calls itself is a recursive function. This method is used when a certain problem is defined in terms of itself. Although this involves iteration, using an iterative approach to solve such a problem can be tedious.

Classes and Objects

Classes are used to create user-defined data structures. Classes define functions called methods, which identify the behaviors and actions that an object created from the class can perform with its data. While the class is the blueprint, an instance is an object that is built from a class and contains real data.

Instantiate an Object in Python

Instance_name=ClassName()

Instance attributes and class attributes

Attributes created in ._init_() are called instance attributes. An instance attribute's value is specific to a particular instance of the class.

On the other hand, class attributes are attributes that have the same value for all class instances. You can define a class attribute by assigning a value to a variable name outside of ._init_().

The self

Class methods must have an extra first parameter in the method definition. We do not give a value for this parameter when we call the method, Python provides it. If we have a method that takes no arguments, then we still have to have one argument.

_init___method

The _init_ method is similar to constructors in C++ and Java. Constructors are used to initializing the object's state.

Inheritance in Python:

Inheritance is the capability of a class to inherit all properties and methods of base class from which it is derived and can add new features to the class without modifying it.

Implementation:

```
class Employee:
    def __init__ (self, designation : str = 'Developer', frontend : bool =
False, backend : bool = False
):
        self.designation = designation
        self.frontend = frontend
        self.backend = backend

    def __repr__ (self):
        return '{} {} {}'.format (self.designation, self.frontend,
self.backend)

    ### Write the your method over here.
    def verifier(self):
        if self.frontend and self.backend:
            return "Fullstack"
        elif self.frontend:
            return "Frontend Developer"
        elif self.backend:
            return "Backend Developer"
        else:
            return "Not a Developer"

if __name__ == '__main__':
    firstEmployee = Employee(frontend=True, backend=True)
    secondEmployee = Employee(frontend=True, backend=False)
    thirdEmployee = Employee(frontend=False, backend=True)
    fourthEmployee = Employee(frontend=False, backend=False)
    print(firstEmployee.verifier())
    print(secondEmployee.verifier())
    print(thirdEmployee.verifier())
    print(fourthEmployee.verifier())
```

```
Fullstack
Frontend Developer
Backend Developer
Not a Developer
```

Conclusion:

Functions and object-oriented programming are core components of Python that promote modularity, code reusability, and clarity. Functions allow programmers to encapsulate blocks of logic that can be reused throughout a program, including recursive functions which solve problems defined in terms of themselves. Object-oriented concepts such as classes and objects provide a blueprint for building real-world models. Using constructors (__init__), class and instance attributes, and the self parameter enables better data management within objects. Inheritance further enhances code reusability by allowing new classes to derive features from existing ones