Experiment No. 4
Creating functions, classes and objects using python
Date of Performance: 06/02/2024
Date of Submission: 27/02/2024

Experiment No. 4

Title: Creating functions, classes and objects using python

Aim: To study and create functions, classes and objects using python

Objective: To introduce functions, classes and objects in python

Theory:

A function is a block of code which only runs when it is called.

You can pass data, known as parameters, into a function.

A function can return data as a result.

A class is a user-defined blueprint or prototype from which objects are created. Classes provide a means of bundling data and functionality together. Creating a new class creates a new type of object, allowing new instances of that type to be made. Each class instance can have attributes attached to it for maintaining its state. Class instances can also have methods (defined by their class) for modifying their state.

To understand the need for creating a class let's consider an example, let's say you wanted to track the number of dogs that may have different attributes like breed, age. If a list is used, the first element could be the dog's breed while the second element could represent its age. Let's suppose there are 100 different dogs, then how would you know which element is supposed to be which? What if you wanted to add other properties to these dogs? This lacks organization and it's the exact need for classes.

Class creates a user-defined data structure, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class. A class is like a blueprint for an object.

Program :

class Student:

```
    def __init__(self,n='',m=0,d=''):
        self.name = n
        self.marks = m
        self.department = d
    def printDetails(self):
        print()
        print("Student Details: ")
        print("Name : ",self.name)
        print("Marks : ",self.marks)
        print("Department : ",self.department)
        print()


name = input("Enter Name  : ")
marks = input("Enter Marks : ")
dep = input("Enter Department : ")
s1 = Student(name,marks,dep)


name = input("Enter Name  : ")
marks = input("Enter Marks : ")
dep = input("Enter Department : ")
s2 = Student(name,marks,dep)
name = input("Enter Name  : ")
marks = input("Enter Marks : ")
```

```
dep = input("Enter Department : ")

s3 = Student(name,marks,dep)


s1.printDetails()

s2.printDetails()

s3.printDetails()
```

Output :

```
PS C:\Users\Prathmesh\Desktop\python> python sample.py
Enter Name  : Raj
Enter Marks : 98
Enter Department : comps
Enter Name  : om
Enter Marks : 35
Enter Department : comps
Enter Name  : Ramesh
Enter Marks : 60
Enter Department : comps

Student Details:
Name :  Raj
Marks :  98
Department :  comps


Student Details:
Name :  om
Marks :  35
Department :  comps


Student Details:
Name :  Ramesh
Marks :  60
Department :  comps

PS C:\Users\Prathmesh\Desktop\python>
```

Conclusion:

Classes in Python define structures for creating objects with specific attributes and behaviors, promoting code organization and reusability. Objects are instances of these classes, representing individual entities within a program. Through inheritance and polymorphism, classes enable the sharing and extension of functionality, fostering efficient and modular code development. Overall, classes and objects form the foundation of object-oriented programming in Python, facilitating the creation of flexible and scalable software solutions.