



Vidyavardhini's College of Engineering & Technology
Department of Computer Engineering

| |
|--|
| Experiment No. 4 |
| Creating functions, classes and objects using python |
| Date of Performance: |
| Date of Submission: |



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.



Class Student Using Functions And Objects

CODE:

```
class Student:
    def StudentDetails(self):
        self.rollno=input("Enter Roll Number : ")
        self.name=input("Enter Name : ")
        self.m3=int(input("Enter M3 marks : "))
        self.ds=int(input("Enter DS marks : "))
        self.dlcoa=int(input("Enter DLCOA marks : "))
        self.cg=int(input("Enter CG marks : "))

    def printResult(self):
        self.percentage=(int)((self.m3+self.ds+self.dlcoa+self.cg)/400*100);
        print(self.rollno,self.name,self.percentage)

s1=Student()
s1.StudentDetails()

print("Result : ")
s1.printResult()
```

OUTPUT:

```
===== RESTART: C:/Vedanti_Degree/SEM_4/SBL_PYTHON/pr4.1.py =====
Enter Roll Number : 66
Enter Name : VR
Enter M3 marks : 32
Enter DS marks : 54
Enter DLCOA marks : 68
Enter CG marks : 70
Result :
66 VR 56
```

Function For Factorial Of A Number

```
def factorial (n):
    if n < 0:
        return 0
    elif n == 0 or n == 1:
        return 1
    else:
        fact = 1
        while (n> 1):
```



```
fact *= n
n -= 1
return fact
```

```
num = int(input("Enter a value:"))
print("Factorial of", num, "is", factorial(num))
```

OUTPUT:

```
===== RESTART: C:/Vedanti_Degree/SEM_4/SBL_PYTHON/pr4.2.py =====
Enter a value:5
Factorial of 5 is 120
```

Function To Check Prime Number

CODE:

```
def is_prime (n):
    if n < 2:
        return False
    for i in range(2, int (n**0.5) + 1):
        if n%i == 0:
            return False
    return True

num=int(input("Enter a value:"))
if is_prime (num):
    print(num, "is a prime number")
else:
    print(num, "is not a prime number")
```

OUTPUT:

```
===== RESTART: C:/Vedanti_Degree/SEM_4/SBL_PYTHON/pr4.3.py =====
Enter a value:5
5 is a prime number

===== RESTART: C:/Vedanti_Degree/SEM_4/SBL_PYTHON/pr4.3.py =====
Enter a value:4
4 is not a prime number
```



CONCLUSION:

In conclusion, creating the functions, classes, and objects in Python is pivotal for building robust, modular, and scalable software solutions. Functions provide a way to encapsulate reusable blocks of code, promoting code readability and reducing redundancy. Classes and objects allow for the creation of custom data types with attributes and methods, facilitating the modeling of real-world entities and the implementation of complex behaviors.