



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



# Vidyavardhini's College of Engineering & Technology

## Department of Computer Engineering

---

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.

1)

**Code:**

```
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
```



# Vidyavardhini's College of Engineering & Technology

## Department of Computer Engineering

# Driver Code

```
num = 5

print("Factorial of",num,"is",

factorial(num))
```

**Output:**

```
File Edit Selection View Go Run Terminal Help
py second yr
EXPLORER
OPEN EDITORS
Welcome
list.py
demo.py
factorial.py
PY SECOND YR
demo.py
factorial.py
list.py
Screenshot (133).png
ss.png
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
[Running] python -u "c:\Users\student\Desktop\py second yr\tempCodeRunnerfile.py"
Factorial of 5 is 120
[Done] exited with code=0 in 0.056 seconds
Code
Ln 2, Col 1 (230 selected) Spaces: 4 UTF-8 CR/LF Python 3.11.4 64-bit Go Live
10:22 AM 2/16/2024
```

2)

**Code:**

```
class Employee:

    def __init__(self, name, age):

        self.name = name

        self.age = age
```



# Vidyavardhini's College of Engineering & Technology

## Department of Computer Engineering

---

```
def details(self):
```

```
    print("Employee Name:", self.name)
```

```
    print("Employee Age:", self.age)
```

```
emp = Employee("Samriddhi", 25)
```

```
emp.details()
```

### Output:

```
[Running] python -u "c:\Users\student\Desktop\py second yr\factorial.py"
Employee Name: Sam
Employee Age: 36

[Done] exited with code=0 in 0.046 seconds

[Running] python -u "c:\Users\student\Desktop\py second yr\factorial.py"
Employee Name: Samriddhi
Employee Age: 25

[Done] exited with code=0 in 0.045 seconds
```

### Conclusion:

Classes object and functions have been implemented.