

exp_1

February 17, 2025

1 Experiment 2: Basic elements of Python

- Name: Anas Muhammmmed Sahil
- Date: 09-01-2025
- Roll Number: 20242AIE0010

```
[4]: # Write a Python program to implement Basic Elements of Python such as  
# Branching, Recursion, Global Variables, Modules, Files, Inheritance,  
# Encapsulation and Information Hiding  
"""  
Lab Record:  
1. Variables  
2. Methods & modules  
3. Encapsulation  
4. Files  
5. Branching  
6. Recursion  
7. Inheritance  
8. Information Hiding  
9. Polymorphism  
"""  
  
def fibonacci(n):  
    # Base cases  
    if n == 0:  
        return 0  
    elif n == 1:  
        return 1  
    # Recursive case  
    else:  
        return fibonacci(n-1) + fibonacci(n-2)  
  
def tail_fact(n, acc=1):  
    # Base case  
    if n == 0:
```

```

        return acc
    # Tail recursive call with an accumulator
    else:
        return tail_fact(n-1, acc * n)

def nontail_fact(n):
    # Base case
    if n == 1:
        return 1
    # Non-tail recursive call because the multiplication happens after the call
    else:
        return n * nontail_fact(n-1)

# Parent class
class Animal:
    def __init__(self, name):
        self.name = name

    def speak(self):
        return f"{self.name} speaks!"

# Child class inheriting from Animal
class Dog(Animal):
    def speak(self):
        return f"{self.name} barks!"

# A Python program to demonstrate inheritance
class Person(object):
    # Constructor
    def __init__(self, name, id):
        self.name = name
        self.id = id

    # To check if this person is an employee
    def Display(self):
        print(self.name, self.id)

class HiddenClass:
    __hiddenVariable = 0

    def add(self, increment):
        self.__hiddenVariable += increment

```

```

        print("Expected error, intended")
        print(self.__hiddenVariable)

def main():
    print(fibonacci(10))
    print(tail_fact(5))
    print(nontail_fact(5))
    dog = Dog("Buddy")
    print(dog.speak())
    emp = Person("Satyam", 7)
    emp.Display()
    myObject = HiddenClass()
    myObject.add(2)
    myObject.add(5)
    print(myObject.__hiddenVariable)

if __name__ == "__main__":
    main()

```

```

55
120
120
Buddy barks!
Satyam 7
Expected error, intended
2
Expected error, intended
7

```

```

-----
AttributeError                                Traceback (most recent call last)
Cell In[4], line 99
     95     print(myObject.__hiddenVariable)
     98 if __name__ == "__main__":
--> 99     main()

Cell In[4], line 95, in main()
     93 myObject.add(2)
     94 myObject.add(5)
--> 95 print(myObject.__hiddenVariable)

AttributeError: 'HiddenClass' object has no attribute '__hiddenVariable'

```