OOPS -- Object Oriented Programming System

These are the 4 pillars of OOPs in Python:
Encapsulation
Abstraction
Inheritance
Polymorphism

1. Class and Object: A class is a blueprint, and an object is an instance of that class.

2. Inheritance: A class can inherit properties and methods from another class.

3. Polymorphism: Same method name, different behaviour.

4. Encapsulation: Restricting access to variables/methods using _protected and __private.

5. Abstraction: Hiding details and showing only essential features (using abc module).

Code:

```
class Calculator:
    def __init__(self,name):
        self.name = name
    def add(self,a,b):
        return a+b
    def sub(self,a,b):
        return a-b
    def mul(self,a,b):
        return a*b
    def div(self,a,b):
        return a/b
    def rem(self,a,b):
        return a%b
```

```
name = input("Enter Your Name: ")
c1 = Calculator(name)

print("Operations:")
print(" 1.Addition\n 2.Substractiopn\n 3.Multiplication\n 4.Division\n
5.Remainder\n 6.Name\n 7.Exit")

while True:
    n=int(input("Select: "))
    if n == 1:
        a,b = map(int,input("Enter values: ").split())
        print("Result: ",c1.add(a,b))

    if n == 2:
        a,b = map(int,input("Enter values: ").split())
        print("Result: ",c1.sub(a,b))

    if n == 3:
        a,b = map(int,input("Enter values: ").split())
        print("Result: ",c1.mul(a,b))

    if n == 4:
        a,b = map(int,input("Enter values: ").split())
        print("Result: ",c1.div(a,b))

    if n == 5:
        a,b = map(int,input("Enter values: ").split())
        print("Result: ",c1.rem(a,b))

    if n == 6:
        print(name)

    if n == 7:
        break
```