Programming Paradigms in Python

- Imperative Programming Paradigm
  - How the program should do something
  - For loops, While loops
- Declarative Programming Paradigm
  - What the program does
  - Map, reduce, SQL queries

Three types of programming paradigms:

- Object Oriented programming paradigms
- Procedure Oriented programming paradigms
- Functional programming paradigms

Object Oriented programming paradigms

- Organizes code using classes and objects. It supports principles like inheritance, encapsulation, and polymorphism.
- code can be easily reusable.

Example:

```
class Calculator:
    def __init__(self, a, b):
        self.a = a
        self.b = b


    def add(self):
        return self.a + self.b
```

```
calc = Calculator(10, 20)
 print("Sum is:", calc.add())
```

Procedural programming paradigms

- Involves writing procedures or functions that perform operations on data. It follows a **top-down approach**.

Example:

```
def add(a, b):
    return a + b

def main():
    x = 10
    y = 20
    print("Sum is:", add(x, y))

main()
```

Functional programming paradigms (Declarative paradigms)

- Treats every statement as functional expression
- Treat functions as values and pass them as an arguments
- Making debugging and testing easier

Example:

```
 numbers = [1, 2, 3, 4]
 squared = list(map(lambda x: x**2, numbers))
 print("Squared numbers:", squared)
```