Python Programming Language Paradigms.

1. Procedural Programming: This paradigm focuses on writing a sequence of instructions to solve a problem. It involves breaking down a program into smaller subroutines or functions that perform specific tasks. In Python, procedural programming is achieved using functions.
2. Object-Oriented Programming (OOP): This paradigm involves modeling the problem domain using objects that have properties (attributes) and behaviors (methods). Python supports OOP features like classes, objects, inheritance, encapsulation, and polymorphism.
3. Functional Programming: This paradigm treats computation as the evaluation of mathematical functions and avoids changing state and mutable data. Python supports functional programming constructs like lambda functions, map, filter, and reduce.
4. Aspect-Oriented Programming (AOP): This paradigm involves separating cross-cutting concerns, such as logging and error handling, from the main program logic. Python supports AOP using libraries like AspectLib and PyAOP.
5. Event-Driven Programming: This paradigm is used in applications that respond to events such as mouse clicks, key presses, and network activity. Python supports event-driven programming using libraries like PyQt and wxPython.

<https://github.com/CharityOkekwu/Assignment-on-paradigms-3-05-2023>Bottom of Form