Procedural & OOP

Procedural programming and Object-Oriented Programming (OOP) are two different programming paradigms that structure and organize code in distinct ways. Here are the key differences between procedural and OOP languages:

Procedural Programming:

1. **Focus**: In procedural programming, the focus is on procedures or functions that perform specific tasks. The program is organized around a sequence of procedures that manipulate data.

2. **Data**: Data is often stored in global variables, and functions operate on this data. Procedures are typically separate from the data they act upon.

3. **Modularity**: Procedural programs are divided into smaller functions, promoting code modularity. Functions can be reused in various parts of the program.

4. **Code Reusability**: Code reusability is achieved through function calls, where functions can be invoked from multiple parts of the program.

5. **Encapsulation**: Data is not encapsulated; it's usually accessible globally or through function parameters.

Object-Oriented Programming (OOP):

1. **Focus**: OOP emphasizes the creation and interaction of objects, which are instances of classes. It's centered around modeling real-world entities and their interactions.

2. **Data**: Data is encapsulated within objects (instances of classes), and objects have methods (functions) that operate on their own data. This promotes data hiding and encapsulation.

3. **Modularity**: OOP promotes a high degree of modularity through classes. Each class encapsulates data and the operations that can be performed on that data.

4. **Code Reusability**: Code reusability is achieved through inheritance, where new classes (subclasses) can inherit attributes and behaviors from existing classes (superclasses).

5. **Encapsulation**: Data is encapsulated within objects, and access to that data is controlled by methods. This helps protect data from unintended modification.

In summary, procedural programming focuses on procedures and functions that manipulate data, while OOP focuses on creating objects that encapsulate data and behavior. OOP promotes code organization around classes, inheritance, and encapsulation, making it well-suited for modeling complex real-world systems and enhancing code maintainability and reusability. Procedural programming is more straightforward and linear, often used for simpler tasks or when performance is a critical concern. The choice between the two depends on the specific needs of a project.