Coding Best Practices-

1. Variable naming conventions
2. Class and function naming conventions
3. Clear and concise comments
4. Indentations
5. Portability
6. Reusability and scalability
7. Testing
8. Re-Usability
9. Security – public/private/protected/default
10. All scenarios including Exceptions

Follow SOLID principles and Design Patterns

* S - Single-Responsibility Principle

A class should have one and only one reason to change, meaning that a class should have only one job.

* - Open-Closed Principle

Objects or entities should be open for extension but closed for modification

* L - Liskov Substitution Principle

Let q(x) be a property provable about objects of x of type T. Then q(y) should be provable for objects y of type S where S is a subtype of T.

* I - Interface Segregation Principle

A client should never be forced to implement an interface that it doesn’t use, or clients shouldn’t be forced to depend on methods they do not use.

* D - Dependency Inversion Principle

Entities must depend on abstractions, not on concretions. It states that the high-level module must not depend on the low-level module, but they should depend on abstractions.

Ref - https://www.digitalocean.com/community/conceptual\_articles/s-o-l-i-d-the-first-five-principles-of-object-oriented-design