Good programming practices for C++

1. **Encapsulation (Private and Public Members):** Encapsulate class data by using access specifiers (private, protected, and public). Keep data members private to control access and use public methods (getters and setters) to manipulate them.

2. **Abstraction:** Hide implementation details and provide a clear interface to the outside world while hiding implementation details.

3. **Use Virtual Functions:** Utilize virtual functions and polymorphism to achieve dynamic behavior. Declare base class functions as virtual and override them in derived classes.

4. **Initialize Members in Constructors:** Ensure all class members are properly initialized in constructors.

5. **Resource Management using Destructor:** Implement proper resource management in destructors, especially for dynamically allocated resources.

6. **Use const Correctly:** Apply const to member functions that do not modify the object's state. Use const references where applicable.

7. **Operator Overloading:** Overload operators when it makes logical sense for your class. Avoid overloading operators in ways that might confuse the code reader.

8. **Descriptive Naming:** Use descriptive and meaningful names for classes, methods, and variables.

9. **Readable Code:** Write clear and readable code. Break down complex functions into smaller, modular functions.

10. **Use Meaningful Comments:** Provide comments where necessary to explain important code sections.

11. **Avoid Global Variables:** Avoid the use of global variables whenever possible. Minimizing the global state improves code maintainability and testability.

12. **Indentation:** Property indent the code, as indentation help in readability, debugging, updating and maintaining the code