1. All design ok, but functions and variables are Boss.
2. If a higher level of absrraction is available , picking up lower level is a good to ensure customer becomes a fossil before we make the product.
   1. Using STL and not writing our own code
   2. Using Async instead of using threads
3. Treat class as a data type
   1. Don’t ignore constructors, destructors and operator overloading function if any
   2. Rule of 5 or rule of 3
4. Namespace usage
5. Remember that picture.
6. Bias towards reference usage.
7. Liberally use const.
8. Use pointers only when needed
   1. Dynamic allocation
   2. Freak cases..
9. Think of using constexpr in case it applies..
   1. Staticassert.
10. Know oops concept especially
    1. Relationship between classes
    2. Interface usage that paintbrush picture.
    3. Remember inheritance and interface have got a checklist
11. Datastructures.
    1. Don’t start with name.
    2. Start with operations
    3. Think which one to pick.
12. Design patterns
    1. Excel sheet first
    2. Class principles next
    3. Then comes design patterns.
13. Templates are useful
    1. Compile time.
    2. Copies
    3. Perfect forwarding..
       1. function has to differentiate lreference and rreference
    4. remember we can put constraints on template types..
       1. concepts (c++20)
       2. typetraits ( applicable to us)
14. bad intent +great process , intent will win
    1. UML
15. Performance
    1. Don’t trust , check.
16. Misc points
    1. Style of coding ok good to know but not a show stopper
       1. Lamda expressions.