Overall summary

1. How to get covert English into functions and variables fitting an architecture.
2. BASIC
   1. Loyalty towards project rather than technology… respect architecture
   2. After all the talk for programmer functions and variables.
   3. Tolerate uncertainty , I will getting doubts.
      1. Lower the cost of failure..
      2. Can I write a small program to clarity my doubt.
3. What does it mean in c++
   1. We have to know what will change, what will not change,
      1. This will decide number of header files.
   2. Models identification..
      1. Classes
   3. Class
      1. Member variables
      2. Member functions
      3. Constructor design
      4. Do you need destructor
      5. Do you need operator overloading.
      6. (another new type) We will come with relationship between classes
         1. Association ( Aggregation and Composition)
         2. Inhertiance.
   4. Functions
      1. Function definition
         1. Input argument
         2. Return type
         3. What does return type mean for the caller.
         4. What type of variable will you use for input and return type
            1. Pointer variable
            2. Reference variable
            3. Normal variable which is not a or b.
         5. If you are inside a class
            1. Static
            2. Non static function
      2. Function calling
         1. Open documentation see
            1. What it does
            2. What does input argument mean
            3. How do you know failure or success
      3. Function assignment.
         1. During assignment we are not calling the function.

Concepts

1. Class
   1. Data type
   2. Encapsulation
2. Static and non static.
   1. Variables
      1. Static
         1. It can exist without object
         2. It can be accessed by using Classname::varaiblename.
            1. Provided the variable is not private.
      2. Non static variable
         1. It is object identity.
         2. Every object will have its own copy
         3. We access it via non static functions typically
   2. functions