

1. There are two audiences for our code. The human reader, and the compiler. The human reader sees that the member variable is private so, they would assume there will be no need to declare the private mem variables, however doing so would prevent the compiler from compiling the program since it needs to proper declaration with the correct inclusion in order to compile.

2. Initializers are preferable because of two reasons,
-- when you construct using assignment statements, you are first initializing the var, then copying the value (string, int,...) into the variable, which means there is an extra step (the cost of copying what you are passing in)
-- Another thing is you run into risk of incorrect construction of that object when you use assignment statements. (Initialzers guard against this by correctly calling the constructor of the object instead of relying on the user to do the 'right' thing when they use assignment statement)

3. We don't want to give the user the ability to mess up the variable and change its value without the change running through us first. That is why we make some member vars private and use the "Setters" to make sure the user does not enter illegal inputs (or input that we are not looking for)