1. We need to declare private variables in header files so that the complier knows what member variables cannot be touched by code from outside of the class. If the complier sees that a function in another source file outside of this class is using a private member variable, it can throw a compile time error to let the programmer know.

2. If a member variable in the class does not have a default constructor to initialize the member variable to some value, the variable cannot be initialized and will throw a compile time error. Therefore initializers are preferred to correctly initialize each member variable.

3. By making the member variable artist private, we can be sure that anyone else using our class Song will not be able to change them in a way that may cause problems. For example, if we have the limit of 30 characters on the artist string while making the variable public, the string may be modified to be longer than 30 characters from an outside function, which may lead to unexpected behavior in other member functions.

4. If we forgot to mark a member function const when it should be, the function will work fine today but leave room for error in the future. If we want to change the code inside of this member function afterwards, we could add code that modifies member variables which should not be allowed and cause more bugs that can be difficult to find.
If we mark a member function const when it should not be, and then proceed to write functions that change member variables of the class, the compilers will throw an error since modifying variables in function with const is not allowed.