1. **Uso del espacio de nombres (`namespace`)**:
    - `cout << "Default Template" << endl;`: Esta línea de código asume que `cout` y `endl` están en el espacio de nombres global, lo que significa que se debe haber usado la instrucción `using namespace std;` anteriormente en el código.
    - `std::cout << "Default Template" << std::endl;`: En esta línea, se especifica explícitamente el espacio de nombres `std` para `cout` y `endl`, lo cual es una buena práctica para evitar confusiones o colisiones de nombres.
2. **Buenas prácticas de programación**:
    - `using namespace std;` puede causar conflictos de nombres si hay funciones, variables o clases con los mismos nombres en otros espacios de nombres. Por eso, es preferible usar `std::cout` y `std::endl` para indicar claramente que estás utilizando la librería estándar de C++.
    - `std::cout << "Default Template" << std::endl;` es más claro y hace el código más legible, especialmente para otros programadores o cuando se trabaja en proyectos grandes.
3. **Desempeño**:
    - En términos de rendimiento, no hay una diferencia significativa. Ambos harán lo mismo porque internamente están usando `cout` y `endl` del mismo espacio de nombres `std`.
4. **Estilo y Mantenimiento**:
    - Usar `std::cout` y `std::endl` sin `using namespace std;` reduce la probabilidad de errores relacionados con el uso no intencionado de nombres similares en otros espacios de nombres, mejorando la mantenibilidad del código.

En resumen, aunque ambas líneas logran lo mismo, `std::cout << "Default Template" << std::endl;` es la opción más recomendada porque sigue las mejores prácticas de programación en C++.