CSE 210-13

W02 Prepare - Articulate

Scott LeFoll

07/14/22

**What is maintainability and why is it important??**

Computer software is by it’s nature very complex, even in the most simple implementations. I know that when we were studying abstraction we learned that the simple python pint() statement utilizes almost 3,000 lines of code to implement. As code requirements expand and as it ages requires modification, the complexity level usually rises substantially. Maintainability refers to the relative cost of updating, debugging and upgrading a code base or its component parts, in terms of time, effort, difficulty and overall cost.

The maintainability of code is sharply increased by following standards, including both industry standards as well as company and team standards, and by following accepted best practices. These especially include the principles of abstraction, polymorphism, inheritance, and encapsulation that we learned in this course. All of these concepts work together to protect the code from needless complexity, redundancy and from the impact of a public interface.

By abstracting, complex use case processes are distilled down to essential ones. Encapsulation protects code, and decreases the modules that would require modification in the case of changes to that code. Inheritance reduces the redundancy of code by sharing inherited methods and attributes, while polymorphism allows an inheriting child class to extend the inherited code by adding its own implementations. All of these things serve to increase the maintainability of the code.