

1. The process of TDD is writing tests before writing any functional code. Then, only writing code to satisfy the failing test. By repeating this over and over, eventually the program will reach full functionality.
2. I feel that these claims are true. When working in very small cases, it becomes easier to see the methods used to write the code, and the programmer seems to have a better feel for what is going wrong when methods fail. I feel that this makes for better code, and coding practices.
3. TDD can be slow, for a relatively simply program taking a TDD approach made the project significantly longer. I imagine this would continue to be the case on larger scale projects. However, it gave me a better idea for what the code was doing in general, and understand why test cases were failing.