1. Software has had immense effects, both negative and positive, on the world we know today. The greatest negative that comes to mind is social media, and the smartphone. Social media dominates the modern cultural and political landscapes. Misinformation is a major issue and is a driven by the super power tech companies. Personal information is taken and sold to said companies and used to take advantage of the human psyche for mere advertising purposes. This is only a small portion of what makes social media and the effects of software so dystopian. However, on a more positive note, software creates unparalleled convenience. For example, we can simply go on our phones and view the weather, or log in to our banks.
2. What a lot of companies do to prevent deterioration of software lies in alpha and beta releases. These releases of software (either released to the public as a whole or a limited number of people who sign up) allow for leeway room for the development team to receive feedback from consumers.
3. A simple yet entirely plausible example of a doomsday software failure could be an accidental nuclear missile launch. While unlikely, such computer failure is still possible.
4. Two more myths about software
   1. Microsoft Windows is a reliable and secure operating system that is useful in many ways.
      * This is NOT true. Microsoft Windows is an absolute dumpster fire of an operating system. It’s insecure, ugly as anything, and is constantly experiencing bugs that have existed since each release. It’s overpriced and unnecessary. Linux is much better and much more reliable.
   2. Computer Programming requires you to remember and simply know everything.
      * This is also NOT true. Computer Programming is extremely collaborative and requires a lot of researching and referencing at all times. Remembering each built-in function of every programming language would be near-impossible.
5. 1. Software rarely ever gets completely “finished,” meaning that updates are typically being constantly released to patch bugs that are encountered by users. However, before software can even be released, developers must assure that every piece of said software functions in accordance to how it is advertised. They cannot afford to deliver false promises
   2. Development costs tend to be high due to the processing power it takes to develop well-designed software. Computers aren’t cheap, and to even test a software requires vary different specs of hardware. This helps assure that the software works as designed with most modern technology.
   3. Software development teams aren’t typically very large, which makes finding small, unique bugs difficult. Upon releasing software to the public, however, hundreds and thousands of people are now able to be testers. As previously mentioned, there’s often unique bugs that only certain users might encounter, and development teams need their feedback in order to discover the source of the problem.
   4. To maintain and existing piece of software is essential in making said piece of software reliable and efficient. Commitment to one single project generates loyalty and respect from consumers. Spending time with the sole focus on a single piece of software allows for it to become extremely polished.
   5. Progress in software development isn’t always linear. Sometimes, features are added that consumers wind up disliking, and pressure is added to move backwards and remove them.