1. **Define what Reliability, Scalability, and Maintainability mean in terms of Data Systems. (2.5 pts)**

Reliability is the ability of a system to continue working despite human error, hardware, and software issues. For human error, it can mean unintentional user inputs or using the program in a way it wasn’t designed for.

Scalability is the design of a system to cope with growth in volume (both in what a user provides as well as the number of users) and complexity of the system. Because systems and the stakeholders involved often evolve, this is an ongoing process that may include system overhaul.

Maintainability is the inherent design of a data system to accommodate new users and different users working on it. Maintainability, according to Kleppmann’s text, relies on three basic tenets: Operability, Simplicity, and Evolvability.

1. **Describe three types of issues that cause problems with Reliability. (2.5 pts)**

*Hardware faults* – a computer part (RAM, hard drive, or power supply, for example) fails and so the system is made unavailable. Software solutions to fault tolerance (distributed system design) and hardware redundancy are common ways to solve hardware faults.

*Software faults* – generally system-wide issues that arise from a process that takes up a shared resource, receives an unexpected input (similar to the fears that Y2K sparked), or cascading failures.

*Human error* – Using systems in unintended ways, providing windows for users to feed the system unintended feedback. A simple example might be a user feeding a phone number field a string of alpha characters – proactive steps can be taken to avoid this, but is a good example of human error.

1. **Describe a few ways we can deal with and plan for Scalability.  (2.5 pts)**
   1. Write code with efficiency in mind – Big O notation can help engineers describe scalable code since its purpose is to describe worst-case runtime for an operation or set of operations. Engineers can also utilize the concept of *loose coupling* to help create development ecosystems that focus on several microservices each performing one thing well, stitched together, as opposed to a monolithic application that must be changed in an all-or-nothing fashion. To that end, engineers can also create sandbox environments to perform extensive QA and user acceptance testing to ensure redesigned applications behave as expected under anticipated load.
2. **Describe the design principles to take into consideration when dealing with Maintainability. (2.5 pts)**  
   When dealing with maintainability, there are three basic considerations to take into account: Operability, Simplicity, and Evolvability. Operability deals with the ability of the operations team to do their job, which includes what happens with the code or applications after deployment (security, performance issues, anticipation of future issues). Simplicity deals with the complexity of the data system, removing unnecessary complexity (in large systems, some level of complexity will be unavoidable). Evolvability deals with the ability of the system to grow and change as the business needs and/or system requirements change (ingesting GB of data to ingesting TB of data may require changing the system in a baseline, structural way, for example).