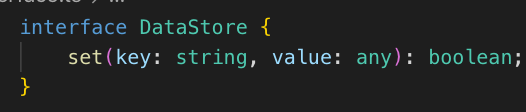
4) Explain the difference between interfaces and implementation.

An interface is the outline or functionality an object must provide to its users; the object must adhere to the interface it implements. An implementation is the internal workings of the object regarding how it works to meet the agreed upon interface. If designed properly, an implementation can be changed from top to bottom, but as long as the object sill meets the requirements of the interface, the user will not be able to tell a difference.

5) Using both visual and written descriptions, think through the interface-implementation of a large scale storage system.   In many systems today, we have the ability to store information from a single application to a variety of storage devices - local storage (hard drive, usb), the cloud and/or some new medium in the future.   How would you design an interface structure such that all of the possible implementations could store data effectively.

When thinking about what the interface would look like for working with data, the CRUD acronym comes to mind (Create, Read, Update, Delete). If I was faced with the requirement of building a class that could store data anywhere, I would focus my interface primarily on having an interface that implemented the CRUD operations. However, since the requirement here is only regarding the storage of data, I would only expose the create functionality to the end user in my public interface. This follows the idea that you should expose the minimal interface possible to the end user; if the user asked for other functionality (which they probably would) in the future, we can add to our interface. It is always easier to add to than remove from an interface.

I would expose a single set method that takes a key (a string representing the name of the data being saved), and the value (the data being saved), which could be any type. I would return a Boolean to indicating if the operating was successful.



The implementation details of how the data is stored could (and most likely would) be vastly different based on the storage medium, but as long as the object the user is interacting with implemented the interface they would not have to change their code to account for differences in storage mediums.