Appendix L …..An Essay on Interfaces

Students often question the usefulness of Interfaces. This essay presents an interesting point of view and, hopefully, shows their true utility.

What is the physical interface to the various systems of an aircraft that a pilot sees when he sits in the cockpit?

The interface consists of all the instruments, dials, gauges, and controls that a pilot sees in front of him. For example, he doesn't deal directly with the elevator at the rear of the plane…rather the yoke (part of the interface) control circuitry that, in turn, moves the elevator and thus makes the plane go up or down.

Now suppose we are the aircraft manufacturer and we have a number of models that we produce. When a pilot sits in the cockpit of **any** of our models, we want him to see essentially the same interface. In other words, we want the same color scheme, things to be essentially in the same position, and the controls to work basically the same…**we want standardization**. Thus a pilot who has flown one of our models could feel fairly comfortable when moving to another model he has never flown.

How do we make sure all our models have this common interface? We send the specification for what the layout of the cockpit is to our design engineers so they will work this basic design into new models they create.

…. And so it is with software interfaces. We tell the software engineers who create classes for us, the signatures of the methods we want. (We do this by giving them an *interface*.) All we have to do is look at the first line of their class and see if it says *implements*. We need not look further. We are assured that they have implemented every method of the interface we specified… otherwise, their class won't compile.

Thus we see that the *interface* does four things for us:

1. It lets us specify the exact method signatures we want in a class that someone else will design for us… without us having to implement the code.
2. It promotes uniformity if several classes implement this same *interface*… just as the airplane cockpit will be uniform between the various models.
3. We can look at the first line of a class and if it says *implements*, we know the author has implemented **everything** we specified… we have no need to look further in the code to be assured of this.
4. Someone who wants to know how to use a class need not look through what might be thousands of lines of code that make up the class. It would be much easier to look at the *interface* to see how to use the class. Let's say that another way. It's **much** easier to look at the *interface* document to see how we **interface** to the class.