**Universidad De Las Fuerzas Armadas - ESPE**

**Members:** Josué Merino, Darian Martinez

**Theme:** Summary of Chapter 6 *“Objects and Data Structures”*

As we do not want to display the details of our data, we require certain in abstract terms, for which it requires to use in a correct and concise way the interfaces "getters" and "setters".

The procedural code makes it difficult in one way to add new data structures since the functions must change in this way. In this way, codes that are difficult for Object Oriented Programming are easy for procedures and vice versa.

In any system there will be times when, as programmers we want to add new data instead of new functions, for when this happens, object-oriented programming is the most opportune way to make such changes. And when we want to change functions instead of data types, the data structures and the procedure will be the most appropriate ways.

**Demeter's law**

A module must not know about the interior of the objects it operates, which means that an object must not show its internal structure through the accessors since, at the time of doing so, we would be exposing and not hiding

**Hybrids**

They are half object and half structure. Its methods do important things, where there are public variables and mutators, private public variables, looking for other external methods to want to use these variables.

Hybrids hinder are not recommended for adding new features or new data, and you should avoid    creating them

**Conclusion**

Objects hide data and display behavior, making it easier to add new objects without changing existing behaviors, making it difficult to add new behaviors to objects. Data structures do not hide data, their behavior is not significant. Thanks to this we can more easily add new behaviors to data structures, but it also makes it difficult for us to add new data structures.

Good software engineers understand these problems and choose the right way to do a job.