

## Week 6: Threaded Discussion II

**From your reading of chapter 8 and your own knowledge from other sources, discuss the following:**

Discuss/Describe the differences between object-oriented and traditional testing.

Robert V. Binder lists some of the reasons Object-Oriented Programming (OOP) presents hazards in addition to those posed by procedural programming:

- Dynamic binding and complex inheritance structures create many opportunities for faults due to unanticipated bindings or misinterpretation of correct usage.
- Interface programming errors are a leading cause of faults in procedural languages. Object-oriented programs typically have many small components and therefore more interfaces. Interface errors are more likely, other things being equal.
- Objects preserve state, but state control (the acceptable sequence of events) is typically distributed over an entire program. State control errors are likely. (Binder, 2005)

OOP's characteristics of modularity, small methods, reuse, and early identification of interfaces make it easier to test; however, inheritance, polymorphism, dynamic binding, complex interfaces, and additional integration make testing OOP systems more complex. For instance, inheritance can require more testing if a class overrides an inherited method or adds new methods (Pfleeger & Atlee, 2006).

### References:

Binder, Robert V (2005). *Testign Object-oriented Systems, Models, Patterns, and Tools*. Addison-Wesley.

Pfleeger, S., & Atlee, J. (2006). *Software Engineering: Theory an Practice*. New Jersey: Prentice Hall.