

## CODE - OBJECT ORIENTED PROGRAMMING

This checklist covers coding issues specific to Object Oriented Programming (OOP) that are not covered in the other *CxCheck\_Code-Xxx* checklists. This checklist does not cover all issues that pertain to OOP, as most of the other coding checklists still apply.

### General

- CDO-1* Do the language and technologies selected for implementing an object oriented design support the OO features the design assumes?
- CDO-2* Is there a clear understanding of how the selected language and technology may affect aspects of how the object model will be constructed, especially in regards to object creation, communication, and persistence?
- CDO-3* Is there a clear understanding of how the static and dynamic aspects of the object model will affect construction, especially in regards to class inheritance, association, aggregation, and delegation?
- CDO-4* Have you created appropriate conventions or other mechanisms to manually provide useful OO support not provided by your language or technology?
- CDO-5* Have consistent patterns, conventions, and idioms been created to ensure consistent implementation of the object oriented design?

### Classes and Interfaces

- CDO-6* Does class construction follow the *CxCheck\_Code-Modules*, *CxCheck\_Code-Routines*, *CxCheck\_Code-Data*, and *CxCheck\_Code-ControlStructures* guidelines?
- CDO-7* Are interface classes kept abstract?
- CDO-8* Are objects created and cleaned up appropriately and at the correct times?
- CDO-9* If appropriate, is reference counting correctly implemented on all resources that need it?

### Inheritance

- CDO-10* If appropriate, is it clear when implementation vs. interface inheritance is being used?
- CDO-11* Have you considered parameterized classes instead of implementation inheritance?

### Polymorphism

- CDO-12* Are type conversions correctly handled with polymorphism either manually or through language or technology support?

### Physical Issues

- CDO-13* Do you fully understand the technology and operational environment your objects will be running in including threading issues, process issues, distributed communication issues, and persistence issues?
- CDO-14* Are you fully aware of technology boundary issues for objects? (e.g., web client context vs. web server context)
- CDO-15* Do you fully understand the implications of late vs. early discovery and binding of the OO technologies you are employing? (e.g., early binding in C++, late binding in Python)