NERSC course syllabus: "OOP in Fortran 2003"

Day 1: Object-Oriented Analysis, Design & Programming

Why Object-Oriented Programming (OOP)?

Conventional programming costs and complexity; Alternative programming paradigms; How performance informs design; How design informs performance.

The Object-Oriented Way

Object-Oriented Analysis (OOA). Object-Oriented Design (OOD). Unified Modeling Language (UML): use case and class diagrams. OOP in Fortran 2003: User-defined structure constructors; encapsulation and information hiding via modules and private scoping; composition, aggregation, and inheritance via extensible derived types; static and dynamic polymorphism via generic interfaces and class variables.

Scientific OOP

User-defined operators and assignments; Abstract data type calculus; Object-oriented design metrics.

Day 2: Best Practices in OOD

OOD patterns: Design patterns essentials and application prototypes; General patterns: Strategy, Surrogate, Abstract Factory, and Factory Method. Domain-specific patterns: Abstract Calculus and Puppeteer. UML object and sequence diagrams.

Day 3: Parallel OOP

Introduction to open-source, object-oriented, parallel Fortran 2003 numerical libraries and interfaces; the ForTrilinos interfaces to the Trilinos solver library and framework; PSBLAS: Design patterns in object-oriented sparse matrix computations. Implementing the Abstract Calculus pattern with Fortran 2008 coarrays and the `do concurrent` construct.

Background Materials

The course draws material from Ref. 1 below. Ref. 2 provides Fortran background. Refs. 3-4 link to open-source software that will be used extensively on Day 3.

1. Rouson, D., J. Xia, and X. Xu (2011) *Scientific Software Design: The Object-Oriented Way*. Cambridge University Press, Cambridge, UK.
2. Metcalf, M., J. Reid, and M. Cohen (2011) *Modern Fortran Explained*. Oxford University Press, Oxford, UK.
3. http://trilinios.sandia.gov/packages/fortrilinos.
4. http://www.ce.uniroma2.it/psblas.

Appendix A of Ref. 1 summarizes useful mathematical material.

Prerequisites

1. Familiarity with Fortran 90 modules, derived types, and kind parameters.
2. Undergraduate-level familiarity with differential equations & matrices.