

TUTORIAL FOR GENERIC PROGRAMMING

TUTORIAL PRESENTATION

Robert O. Jarman
Augusta State University
Augusta, Georgia
rjarman@aug.edu

Howard E. Hinnant
Metrowerks Corporation
Austin, Texas
hinnant@metrowerks.com

Until recently ‘software reuse’, often the mantra of object-oriented programming, has been long on promise and short on realization. The STL, designed by Alexander Stepanov, through its container classes, algorithms, and functions, has evolved into a tool for promoting reuse. The STL is not alone in facilitating reuse, but distinguishes itself from other works in the “high degree of adaptability and efficiency of the components” (Musser, 1996).

This tutorial will introduce the STL, explore its components, and provide representative examples of the STL in action. The tutorial topics will include:

- A brief review of templates

- A description of ‘generic programming’

- STL Components

 - Iterators

 - Container Classes

 - Generic Algorithms

- The STL at work

- The STL in the classroom

 - Abstraction & ADTs

 - Algorithm analysis

 - Software design & problem decomposition

Software engineering, programmer productivity, & metrics

STL Resources

The discussion will be directed toward an audience having a general familiarity with C++ or another OO programming language. Extensive knowledge of C++, templates, or the STL is not assumed or necessary. The tutorial will be “hands-on” providing attendees an opportunity to explore the STL firsthand.

Our goal in this tutorial is to:

Create an awareness of the STL

Provide a modicum of understanding regarding the STL's interface and use.

Give attendees an appreciation of its capabilities

Create an awareness of the potential for instructional use of the STL to augment traditional data structures.

Musser, David R. & Saini, Atul. *STL Tutorial and Reference Guide*, Addison Wesley, 1996