An Idiot's Guide to C++ Templates - Part 1

Alex Sherbakov
Joined 27 Jan 2010
Office Worker
United States

Article Rated: ★★★★★ (Votes: 19, Avg: 5.00)
Posted 27 Jan 2010 · 474,424 views
Updated 27 Jan 2010 · 2 comments

Learning the ins and outs of C++ templates.

Prologue

Back in C++ programming days long ago, C++ templates did not exist. Programmers had to get around this problem. This was a pain and complicated:

- inserting code generation steps into the build process
- complicated macros and magic, and very bugs
- inline assembly code
- cut-and-paste

Fortunately, the C++ standard committee added templates to the language. Now, we can write generic code, without the hassle and pain. Such coding techniques are called generic programming. There is a lot more than just templates to C++ (classes, RTTI, etc.) but templates are a major part of the language.

This article is a tutorial on how to use templates. It is not a complete reference. It is not a tutorial on how to write generic code. It is not a tutorial on how to write template libraries. There are other articles on those topics. This is a tutorial on how to use templates.

Templates are a complex topic. Because of the complexity, I'm breaking this into several articles. Initially, I wanted to cover the entire topic in one long article. I decided against that because it would be too long and too complex. Instead, I'm breaking it into several articles. This first article will cover the basics of templates. Later articles will cover more advanced stuff. Interesting stuff such as template metaprogramming, and tricks and hacks, will not be covered in this series.

Table of Contents

- The Syntax Chaos
- Function Templates
  - Basics
  - Without Templates and Using Void Pointers
  - Multiple Types with Function Templates
  - Function Template Instantiation
  - Function Template Arguments
  - Function Template Overloading
  - Function Template Specialization
- Class Templates
  - Basics
  - Multiple Types with Class Templates
  - Member Function Implementation
  - Class Template Arguments
  - Class Template Specialization
  - Partial Class Template Specialization
- Class Template Overloading
- Class Template Instantiation
- Class Template Arguments
-