

```

/* The following code example is taken from the book
 * "The C++ Standard Library – A Tutorial and Reference, 2nd Edition"
 * by Nicolai M. Josuttis, Addison-Wesley, 2012
 *
 * (C) Copyright Nicolai M. Josuttis 2012.
 * Permission to copy, use, modify, sell and distribute this software
 * is granted provided this copyright notice appears in all copies.
 * This software is provided "as is" without express or implied
 * warranty, and with no claim as to its suitability for any purpose.
 */
#include <functional>

// from boost (functional/hash):
// see http://www.boost.org/doc/libs/1\_35\_0/doc/html/hash/combine.html
template <typename T>
inline void hash_combine (std::size_t& seed, const T& val)
{
    seed ^= std::hash<T>() (val) + 0x9e3779b9 + (seed<<6) + (seed>>2);
}

// auxiliary generic functions to create a hash value using a seed
template <typename T>
inline void hash_val (std::size_t& seed, const T& val)
{
    hash_combine(seed, val);
}
template <typename T, typename... Types>
inline void hash_val (std::size_t& seed,
                     const T& val, const Types&... args)
{
    hash_combine(seed, val);
    hash_val(seed, args...);
}

// auxiliary generic function to create a hash value out of a heterogeneous list
// of arguments
template <typename... Types>
inline std::size_t hash_val (const Types&... args)
{
    std::size_t seed = 0;
    hash_val (seed, args...);
    return seed;
}

```