Fighting Compiler warnings

Marshall Clow

Qualcomm Technology, Inc.

May 18, 2017

The back story

Back in 2012, I implemented simple hex-unhex routines for Boost.Algorithm.

At the bottom of the library, there was a simple routine to convert a 'hex character' to a number.

This is the story of that routine.

Version 0

```
unsigned hex_char_to_int ( char c ) {
      if ( c >= '0' && c <= '9' )
           return c - '0';
      if ( c >= 'A' && c <= 'F' )
           return c - 'A' + 10;
      if ( c >= 'a' && c <= 'f' )
           return c - 'a' + 10;
      BOOST_THROW_EXCEPTION (non_hex_input (c));
      }
```

Not so fast, monkey boy!

```
warning:  control reaches the end of a non-void function.
```

Version 1

```
unsigned hex_char_to_int ( char c ) {
      if ( c >= '0' && c <= '9' )
           return c - '0';
      if ( c >= 'A' && c <= 'F' )
           return c - 'A' + 10;
      if ( c >= 'a' && c <= 'f' )
           return c - 'a' + 10;
      BOOST_THROW_EXCEPTION (non_hex_input (c));
      return 0;    // keep dumb compilers happy
      }
```

As if!

```
warning:  unreachable code in hex_char_to_int.
```

Version 3

```
template <typename T>
unsigned char hex_char_to_int ( T val ) {
    char c = static_cast<char> ( val );
    unsigned retval = 0;
    if      ( c >= '0' && c <= '9' )
        retval = c - '0';
    else if ( c >= 'A' && c <= 'F' )
        retval = c - 'A' + 10;
    else if ( c >= 'a' && c <= 'f' )
        retval = c - 'a' + 10;
    else
        BOOST_THROW_EXCEPTION (non_hex_input()
                                << bad_char (c));
    return retval;
    }
```

Version 4

```
template <typename T>
unsigned char hex_char_to_int ( T val ) {
    char c = static_cast<char> ( val );
    unsigned retval = 0;
    if       ( c >= '0' && c <= '9' )
        retval = c - '0';
    else if ( c >= 'A' && c <= 'F' )
        retval = c - 'A' + 10;
    else if ( c >= 'a' && c <= 'f' )
        retval = c - 'a' + 10;
    else BOOST_THROW_EXCEPTION (non_hex_input()
                                    << bad_char (c));
    return static_cast<char>(retval);
    }
```

And then I got this patch attached to a bug report:

```
template <typename T>
T hex_char_to_int ( T c ) {
    T r;
    if       ( c >= static_cast<T>('0')
            && c <= static_cast<T>('9') )
        r = c - static_cast<T>('0');
    else if ( c >= static_cast<T>('A')
            && c <= static_cast<T>('F') )
        r = c - static_cast<T>('A') + 10;
    else if ( c >= static_cast<T>('a')
            && c <= static_cast<T>('f') )
        r = c - static_cast<T>('a') + 10;
    else BOOST_THROW_EXCEPTION (non_hex_input()
                                << bad_char (c));
    return r;
    }
```