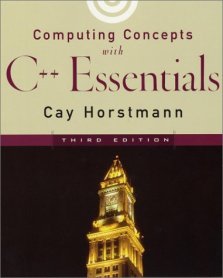
**Welcome to the Big C++/Computing Concepts with C++ Essentials FAQ!**



***Question:* What is the difference between Big C++ and Computing Concepts with C++ Essentials?**

*Answer:*

* The first 16 chapters, and the chapter on GUI programming (chapter 27 of Big C++ = chapter 18 of C++ Essentials) are identical.
* Big C++ has expanded C++ coverage on operator overloading, memory management, exception handling, name scope management, polymorphism, multiple inheritance, templates, and STL (8 chapters). C++ Essentials has a one-chapter summary that briefly discusses operator overloading, exception handling, name scope management, and templates.
* Big C++ has chapters on UML, design patterns, database programming, and XML. (The XML chapter is available on the web.)

***Question:* What are the differences between the second and third edition of Computing Concepts with C++ Essentials?**

*Answer:*

* The coverage of the third edition is more object-oriented, with class design appearing earlier and a chapter on OO design
* Recursion is now covered in a separate chapter
* The chapter on data structures has expanded coverage of STL
* There are new chapters on advanced C++ topics and GUI programming
* There is expanded coverage of pointers and C-style arrays

***Question:* How can I get the solutions to the exercises?**

*Answer:* [Answers to the odd-numbered exercises](http://www.horstmann.com/bigcpp/solutions/solutions.html) are available for all readers. If you are an instructor, please visit <http://www.wiley.com/college/horstmann> and select "Instructor Resources". You will need to fill out a form and obtain a password to see the solutions to all exercises.

***Question:* Which compiler should I use?**

*Answer:* The following compilers should work:

* g++ version 2.95 and later
* Borland C++ 5.5 and later
* Microsoft Visual C++ 6 and later. (If you use Visual C++ 6, be sure to install the [latest patches](http://msdn.microsoft.com/visualc/downloads/vc6.asp) after installing from the CD ROM. The CD ROM version is very buggy.)

The following compilers will definitely not work:

* Microsoft Visual C++ 5 or earlier
* Turbo C++
* Borland C++ 4 or earlier

***Question:* Why do I get an error message: "std does not exist or is not a namespace"?**

*Answer:* Your compiler (Microsoft Visual C++ 6) does not conform to the C++ standard. A remedy is to add a line

namespace std {}

above the using namespace std; directive.

***Question:* Why do I get an error message: "redefinition: multiple initialization" when I have two separate for loops with the same index variable?**

*Answer:* Your compiler (Microsoft Visual C++ 6) does not conform to the C++ standard. A remedy is to rename the index variable in the second loop.

***Question:* Why do I get an error message: "max undefined" even though I include the <algorithm> header?**

*Answer:* Your compiler (Microsoft Visual C++ 6 or g++ 2.9x) does not conform to the C++ standard. A remedy is to add the line

double max(double x, double y) { if (x > y) return x; else return y; }

**Question: Why do the matrix1.cpp and matrix2.cpp programs not compile?**

Answer: Your compiler (Microsoft Visual C++) does not conform to the C++ standard. A remedy is to replace  
  
   static const int ROWS = 3;  
   static const int COLUMNS = 3;  
  
with  
  
   enum { ROWS = 3, COLUMNS = 3 };

***Question:* Why do I get an error message: "no sstream header"?**

*Answer:* Your compiler (g++ 2.9x) does not conform to the C++ standard. A remedy is to change

#include <sstream>

. . .

istr**ing**stream instr(s);

. . .

ostr**ing**stream outstr;

. . .

s = outstr.str();

to

#include <s**tr**stream>

. . .

istrstream instr(s**.c\_str(**))

. . .

ostrstream outstr;

. . .

s = **string(**outstr.str()**)**;

***Question:* Why do I get an error message: "fixed undefined" even though I include the <iomanip> header?**

*Answer:* Your compiler (g++ 2.9x) does not conform to the C++ standard. A remedy is to replace fixed with setiosflags(ios::fixed)

***Question:* Why don't you use the M\_PI constant from the <cmath> header?**

*Answer:* It is not a part of the ISO standard, and some compilers don't define it. If you find it implausible that the standard doesn't define it, you can [purchase an official copy](http://webstore.ansi.org/ansidocstore/product.asp?sku=INCITS%2FISO%2FIEC+9899%2D1999) or [check out an inofficial working draft](http://www.open-std.org/jtc1/sc22/wg14/).

***Question:* Why do you teach doubly-linked lists in the chapter on data structures? Aren't singly linked lists easier?**

*Answer:* (1) STL uses doubly-linked lists. (2) It is actually easier to implement insertion and deletion in a doubly-linked list.

***Question:* Why doesn't the book teach more about char\* strings? Doesn't every C++ programmer need to know them?**

*Answer:* This is a book about *computing concepts*, not about C++. Strings are a *concept*. ANSI C++ supports two *implementations* of strings: the string class and char\* pointers. There is no doubt that many C++ programmers will need to learn both implementations, but I do not believe they should learn all details of both of them in their first programming course. The string class is safe and convenient. Students master it quickly and can move on to learning more computing concepts.

***Question:* My compiler doesn't support #include <iostream> and using namespace std. What is happening?**

*Answer:* The <iostream> header and the std namespace were introduced in 1996 and approved in the international standard in 1998. If your compiler does not support these constructs, you will need to upgrade your compiler. g++, Borland C++ 5.5 and Microsoft Visual C++ 6 are reasonably standard compliant.

***Question:* What platforms does the CCC graphics library support?**

*Answer:* Here is a list of currently supported platforms.

* Any system, with "ASCII art" (#define CCC\_ASC)
* Windows 95/NT (i.e. 32-bit Windows)
* Unix with X11 (uses Xlib only and runs with any window manager)
* Any system that runs [wxWidgets](http://www.wxwidgets.org)

***Question:* I'd like to display color, but all your shapes show up in black only.**

*Answer:* The CCC graphics library has been purposefully kept simple so that students don't fritter away endless time with color and fancy fonts. Use wxWidgets if you want fancier graphics

***Question:* Why do you call an accessors get\_seconds() instead of just seconds()?**

*Answer:* There are many different schemes to name accessors, mutators and data fields. The C++ library uses overloaded pairsseconds() and seconds(int) for accessors and mutators, which I think is a bit too confusing. I felt the get/set terminology makes it really clear that the accessor is a function call. And, of course, that is the convention used in Java.

***Question:* Why do you use all those underscores in function names? What is wrong with mixed case?**

*Answer:* The standard C++ library uses no uppercase letters at all, and it uses underscores to make names more readable (bad\_cast, push\_back). There is nothing wrong with mixed case (getSeconds, readInt); I just wanted to be consistent.