



## Rough course outline

- The basics
  - types, variables, strings, computations
  - simple IO, error handling (if any), exceptions, references, enums, overloading, etc.
  - small C++ code samples
- Data structures and algorithms
  - free store, pointers, and built-in arrays (directly from C)
  - user-defined types (classes)
  - on implementing vectors, lists, iterators..
- Also, safety and resource handling, templates, object-oriented programming & class hierarchies

28.10.2014

Juha Vihavainen / University of Helsinki

3

## A textbook Programming - Principles & Practice with C++ (2. ed.) [Stroustrup, 2014; ~1300 pages] Ch. 1 Computers, People, and Programming Ch. 16 Graphical User Interfaces Ch. 2 Hello,World! Ch. 17 Vector and Free Store Ch. 3 Objects, Types, and Values Ch. 18 Vectors and Arrays Ch. 4 Computation Ch. 19 Vector, Templates, and Exceptions Ch. 5 Errors Ch. 20 Containers and Iterators Ch. 6 Writing a Program (Calculator) Ch. 21 Algorithms and Maps Ch. 7 Completing a Program Ch. 22 Ideals and History Ch. 8 Technicalities: Functions, etc Ch. 23 Text Manipulation Ch. 9 Technicalities: Classes, etc. Ch. 24 Numerics Ch. 10 Input and Output Streams Ch. 25 Embedded Systems Programming Ch. 11 Customizing Input and Output Ch. 26 Testing Ch. 12 A Display Model Ch. 27 The C Programming Language Ch. 13 Graphics Classes App A Language Summary Ch. 14 Graphics Class Design App B Standard Library Summary Ch. 15 Graphing Functions and Data App C - E Visual Studio, FLTK, GUI code Note the emphasis. 28.10.2014 Juha Vihavainen / University of Helsinki 4













## "C++11": even more features

- E.g., auto-typed variables, static assertions, scoped enums, lambdas, rvalue references, delegating constructors, defaulted and deleted functions, new function declarator syntax, inline namespaces, extern templates, local and unnamed types as template arguments, variadic template parameters, thread support library, std::initializer\_list <T>,the "right angle brackets" problem, raw string literals, ..
  - some of these we will discuss as need arises
- "Every feature is implemented somewhere"
- New standard library components are shipping widely
  - e.g. GCC, Microsoft, Boost

28.10.2014

Juha Vihavainen / University of Helsinki

11









<i>II modified for a "console mode":</i>	
#include <iostream></iostream>	<i>   get the required IO facilitie</i>
<pre>int main () {     std::cout &lt;&lt; "Hello, world!\n";</pre>	<ul> <li>// where a C++ program star</li> <li>// output the 13 characters,</li> <li>// followed by a new line</li> </ul>
char c; std::cin >> c; return 0;	<ul><li><i>II wait for an input character</i></li><li><i>II a value indicating success</i></li></ul>
}	
<pre>// without std::cin &gt;&gt; c; the output w // have a chance to read the output (o</pre>	vindow may be closed before you n some implementations)



















Types and literals	
<ul> <li>Built-in types</li> <li>Boolean type <ul> <li>bool</li> </ul> </li> <li>Character types <ul> <li>char, char16_t, char32_t</li> </ul> </li> <li>Integer types <ul> <li>int, short, long, long long</li> </ul> </li> <li>Floating-point types <ul> <li>float, double, long double</li> </ul> </li> </ul>	<ul> <li>boolean literals <ul> <li>true false</li> </ul> </li> <li>character literals <ul> <li>'a', 'x', '4', '\n', '\$'</li> </ul> </li> <li>integer literals <ul> <li>0, 1, 123, -6, 0x34, 0xa3, 1234576L, 123456789LL</li> </ul> </li> <li>floating point literals <ul> <li>1.2, 13.345, .3, -0.54, 1.2e3, .3F, .3F, 13.345L</li> </ul> </li> </ul>
<ul> <li>Standard-library types</li> <li>std::string</li> <li>std::complex <scalartype></scalartype></li> <li>Most types are "standalone"</li> </ul>	<ul> <li>string literals "asdf" (C-style string)</li> <li>complex "literals" (constructor calls)</li> <li>complex <double> (12.3, 99)</double></li> <li>complex <float> (1.3F)</float></li> <li>(1.2, 2.0)</li> <li>(1.4, 2.2, 0.0)</li> </ul>
28.10.2014 Juha Vihavainen / L	Jniversity of Helsinki 26

















