# **Overview of C++: Design Goal Conflicts**

#### Douglas C. Schmidt <u>d.schmidt@vanderbilt.edu</u> www.dre.vanderbilt.edu/~schmidt



#### **Professor of Computer Science**

Institute for Software Integrated Systems

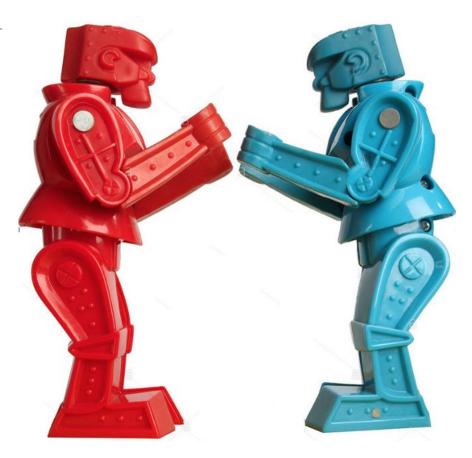
Vanderbilt University Nashville, Tennessee, USA





## Learning Objectives in this Part of the Lesson

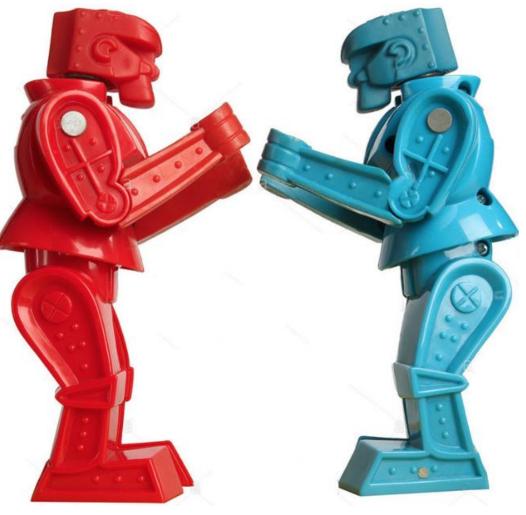
- Recognize the key components of C++
- Know strategies for learning C++
- Understand C++ design goals
- Learn about conflicts of C++ design goals





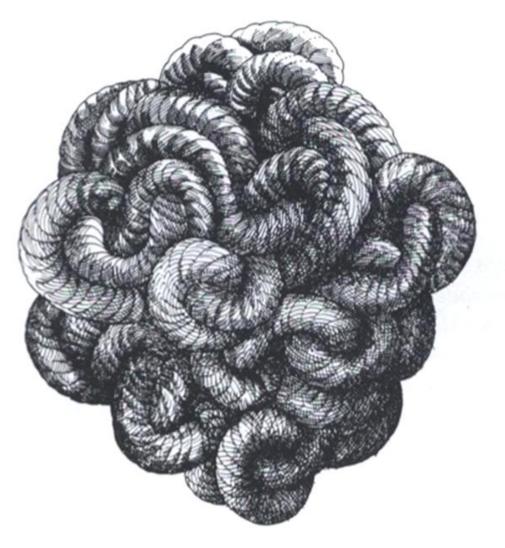


 Certain C++ design goals conflict w/modern techniques & tools



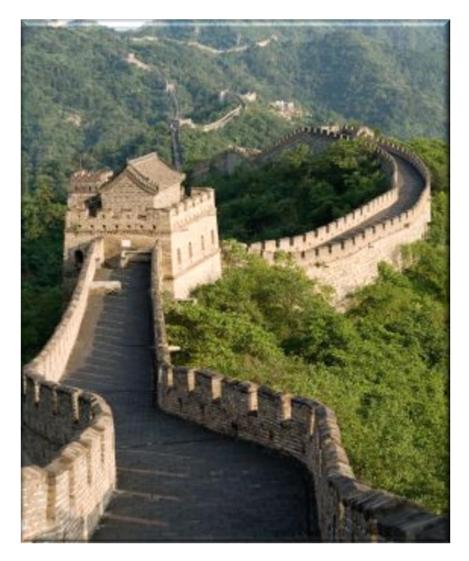


- Certain C++ design goals conflict w/modern techniques & tools, e.g.
  - Compiler optimization
    - Pointers to arbitrary memory locations complicate register allocation & garbage collection





- Certain C++ design goals conflict w/modern techniques & tools, e.g.
  - Compiler optimization
    - Pointers to arbitrary memory locations complicate register allocation & garbage collection
    - Separate compilation complicates inlining due to difficulty of interprocedural analysis





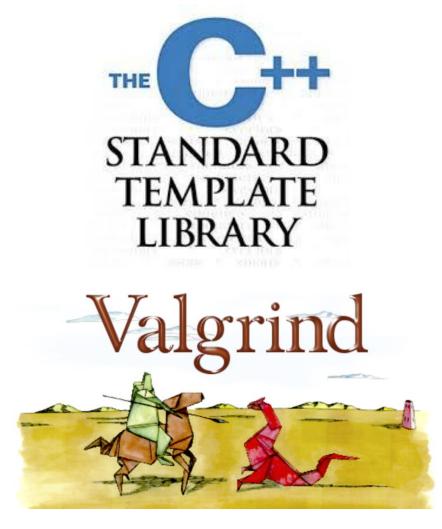
- Certain C++ design goals conflict w/modern techniques & tools, e.g.
  - Compiler optimization
  - Software quality assurance
    - Dynamic memory management & pointers are error-prone

	9016A950,0x00000001,0x00000065) 578 80168950 has base at 80100000
.6.2 irqliif S	YSVER Gx10000565
Nome	Dil Base DateStmp - Name
ntoskrnl.exe	60010000 33247180 a1.d11
atapi.sys	80007000 3324804 SIPORT
Disk.sys	801db000 336015
Ntfs.sys	80237000 344meb4
NTice.sys	fif46000 Blec6c8d Loppy.8
drom.SYS	£228a000 31ea6c9k ull.SYS
KSecDD. SYS	12290000 335 SPA
win32k.sys	1002000 34
Cdfa.SYS	fdda2000 3 S
nbr.sys	£de35000
netht, sys	11168000
ard.aya	£2000000
Parport.SYS	10014000
Parportion ava	r1dd0000

IF HE HAD A NICKLE FOR EVERY TIME WINDOWS CRASHES... ... Oh, wait, never mind



- Certain C++ design goals conflict w/modern techniques & tools, e.g.
  - Compiler optimization
  - Software quality assurance
    - Dynamic memory management & pointers are error-prone
      - Largely fixed in (best) practice





- Certain C++ design goals conflict w/modern techniques & tools, e.g.
  - Compiler optimization
  - Software quality assurance
    - Dynamic memory management & pointers are error-prone
      - Largely fixed in practice, e.g.,
        - Using "resource acquisition is initialization" idiom & "holder" classes

std::ofstream file
 ("example.txt");
if (!file.is\_open())
 throw runtime\_error("...");

file << message << std::endl;</pre>

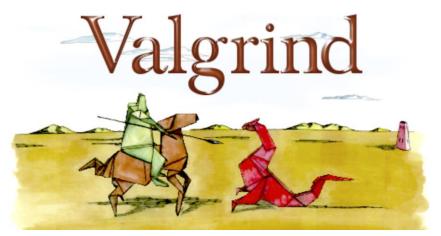
```
// file will be closed
// regardless of exception
// mutex will be unlocked
// regardless of exception.
```

See en.wikipedia.org/wiki/Resource\_acquisition\_is\_initialization

}



- Certain C++ design goals conflict w/modern techniques & tools, e.g.
  - Compiler optimization
  - Software quality assurance
    - Dynamic memory management & pointers are error-prone
      - Largely fixed in practice, e.g.,
        - Using "resource acquisition is initialization" idiom & "holder" classes
        - Memory checking tools



HEAP SUMMARY: in use at exit: 1,000 bytes in 1 blocks total heap usage: 7 allocs, 6 frees, 78,997 bytes allocated LEAK SUMMARY: definitely lost: 1,000 bytes in 1 blocks indirectly lost: 0 bytes in 0 blocks possibly lost: 0 bytes in 0 blocks still reachable: 0 bytes in 0 blocks suppressed: 0 bytes in 0 blocks

# End of C++ Design Goal Conflicts