

Evolution of Program Abstraction Mechanisms: Generic Programming

Douglas C. Schmidt

d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt



Professor of Computer Science

**Institute for Software
Integrated Systems**

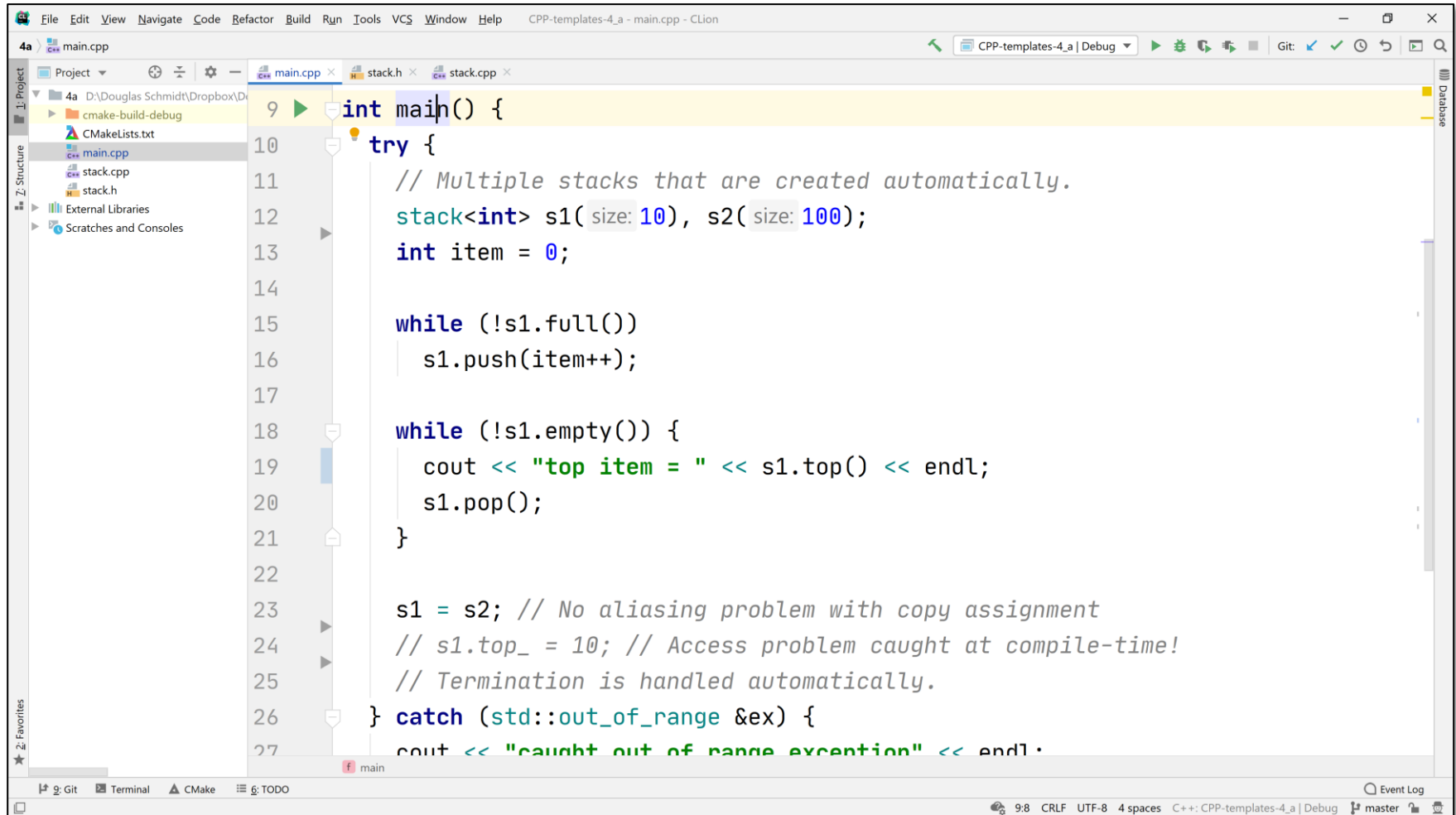
**Vanderbilt University
Nashville, Tennessee, USA**



C++ Generic Programming Stack Implementation

Template Implementation in C++

- A parameterized type Stack class interface using C++



```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help CPP-templates-4_a - main.cpp - CLion
4a main.cpp
Project
4a D:\Douglas Schmidt\Dropbox(D...
  cmake-build-debug
  CMakeLists.txt
  main.cpp
  stack.cpp
  stack.h
  External Libraries
  Scratches and Consoles
Z: Structure
2: Favorites
9 int main() {
10     try {
11         // Multiple stacks that are created automatically.
12         stack<int> s1(size: 10), s2(size: 100);
13         int item = 0;
14
15         while (!s1.full())
16             s1.push(item++);
17
18         while (!s1.empty()) {
19             cout << "top item = " << s1.top() << endl;
20             s1.pop();
21         }
22
23         s1 = s2; // No aliasing problem with copy assignment
24         // s1.top_ = 10; // Access problem caught at compile-time!
25         // Termination is handled automatically.
26     } catch (std::out_of_range &ex) {
27         cout << "caught out of range exception" << endl;

```

See [CPlusPlus/tree/master/overview/4-C++-templates](https://github.com/Douglas-Schmidt/CPlusPlus/tree/master/overview/4-C++-templates)