

# **Evolution of Program Abstraction Mechanisms: Generic Programming**

**Douglas C. Schmidt**

[d.schmidt@vanderbilt.edu](mailto:d.schmidt@vanderbilt.edu)

[www.dre.vanderbilt.edu/~schmidt](http://www.dre.vanderbilt.edu/~schmidt)



**Professor of Computer Science**

**Institute for Software  
Integrated Systems**

**Vanderbilt University  
Nashville, Tennessee, USA**

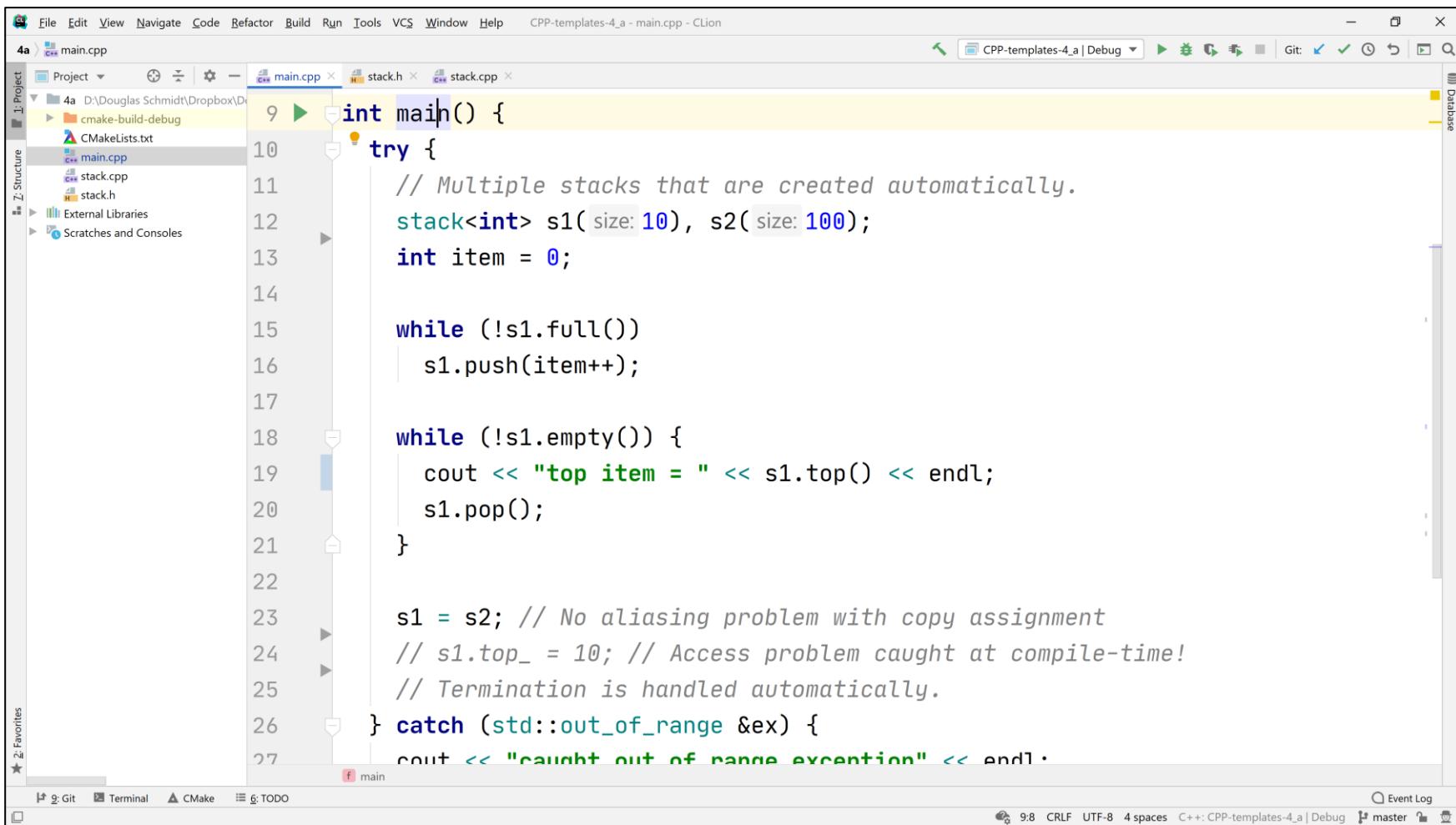


---

# C++ Generic Programming Stack Implementation

# Template Implementation in C++

- A parameterized type Stack class interface using C++



The screenshot shows the CLion IDE interface with the following details:

- File Menu:** File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, Help.
- Project Tab:** 4a, main.cpp.
- Toolbars:** Standard toolbar with icons for file operations, run, build, and Git.
- Code Editor:** Displays the main.cpp file content. The code implements a stack using templates and exception handling.
- Toolbars:** Includes a Database toolbar on the right.
- Bottom Status Bar:** Shows the current time (9:8), file encoding (CRLF), character width (UTF-8), spaces (4 spaces), the current file (C++: CPP-templates-4\_a | Debug), and the current branch (master).

```
int main() {
    try {
        // Multiple stacks that are created automatically.
        stack<int> s1(size: 10), s2(size: 100);
        int item = 0;

        while (!s1.full())
            s1.push(item++);

        while (!s1.empty()) {
            cout << "top item = " << s1.top() << endl;
            s1.pop();
        }

        s1 = s2; // No aliasing problem with copy assignment
        // s1.top_ = 10; // Access problem caught at compile-time!
        // Termination is handled automatically.
    } catch (std::out_of_range &ex) {
        cout << "caught out of range exception" << endl;
    }
}
```

See [CPlusPlus/tree/master/overview/4-C++-templates](https://CPlusPlus/tree/master/overview/4-C++-templates)