Practical Tips for Safer C++

Tristan Brindle

© 75 mins beginner

intermediate

11:00-12:15, Friday, 5th July 2024



In this talk, we'll look at some common safety problems that can occur in everyday C++ code and offer practical advice and suggestions for detecting and avoiding them.

While C++ isn't going to become "a safe language" any time soon, we can certainly make it safer for everyday use -without harming performance. For practical, take-away tips on how you can do so, please join us in this talk!



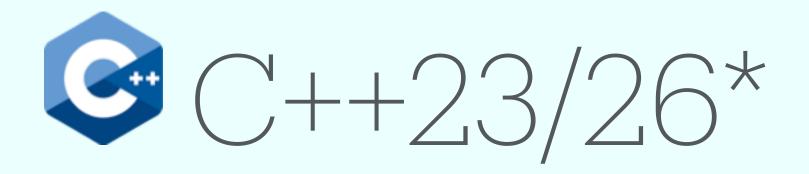
Tristan Brindle

Tristan Brindle is a C++ consultant and trainer based in London. With over 15 years C++experience, he started his career working in high-performance computing in the oil industry in Australia before returning home to his native UK in 2017. He is an active member of the ISO C++ Standards Committee (WG21) and the BSI C++ Panel. He is a regular speaker at C++ conferences around the world, and was formerly a director of C++ London Uni, a nonprofit organisation offering free introductory programming classes in London and online.









• Bounds:

- Use the **flux** library (index based ranges) or **std::ranges**
- Use hardened std library (_LIBCPP_HARDENING_MODE_DEBUG/FAST=1)

• Lifetime:

Static analyser

Initialisation:

Static analyser

• Arithmetic:

- Saturating numeric operations (C++26)
- Use -ftrapv to generate traps for signed integer overflow
- Thread: Tsan