

Index

R

- range expressions
 - lifetime of temporary objects, 691–696
 - range-based **for** loops, 680
- range generators, 687–690
- range-based **for** loops, 571–572
 - annoyances, 703–709
 - adapter requirements, 706
 - argument-dependent lookup (ADL), 707–709
 - sentinel iterator types, lack of support, 706–707
 - state of iteration, lack of access, 703–706
 - description of, 679–684
 - further reading for, 709
 - potential pitfalls, 691–703
 - differences in simple and reference-proxy behaviors, 700–703
 - inadvertent element copying, 696–700
 - lifetime of temporary objects, 691–696
 - specification, 680–683
 - traversing arrays and initializer lists, 683–684
 - use cases, 684–691
 - iterating all container elements, 684–685
 - iterating simple values, 690–691
 - range generators, 687–690
 - subranges, 686–687
- Ranges Library, 391–393, 686n4, 687n5
- raw string literals
 - collisions, 109–111
 - description of, 108–111
 - potential pitfalls, 112–114
 - encoding new lines and whitespace, 113–114
 - unexpected indentation, 112–113
 - use cases, 111–112
- raw UDL operators, 841, 845–849, 870
- reachable, 712
- reaching scope, 587–588
- read-copy-update (RCU) synchronization mechanism, 999
- recursion, 604–605, 875
- recursive initialization, 77–78, 163–165
- recursive lambdas, 977–979
- reducing code size, 1101–1111, 1143–1144
- redundant check, 115
- refactoring with curiously recurring template pattern (CRTP), 1042–1044
- reference collapsing, 380–382
- reference related, 726
- reference types
 - alignof** operator, 184
 - deducing, 198
 - `gs1::span`, 17
 - as literal types, 279
 - `memcpy` usage on, 489–493
 - overloading, 727–730
 - union membership and, 1174
- references, **noexcept** and, 1089–1091
- reflection, 520n46
- ref-qualified, 1154
- ref-qualified overloads, 1171–1172
- ref-qualifiers
 - annoyances, 1171–1172
 - description of, 1153–1160
 - forwarding references, 380
 - further reading for, 1173
 - potential pitfalls, 1170–1171
 - syntax and restrictions, 1157–1160
 - use cases, 1160–1170
 - forbidding *lvalue* operations, 1165–1167
 - forbidding *rvalue*-modifying operations, 1163–1165
 - optimizing immutable types and builder classes, 1167–1170
 - returning *rvalue* subobjects, 1160–1163
- register** keyword, 195n1
- regular types, ~~187n2~~, 751. *See also* types
- reinterpret_cast** keyword, 506–519
- relaxed restrictions on **constexpr** functions, 959–967. *See also* **constexpr** variables; variadic templates
 - description of, 959–960
 - further reading for, 965
 - optimized C++11 example algorithms, 965–967
 - use cases, 961–964
 - nonrecursive **constexpr** algorithms, 961–962
 - optimized metaprogramming algorithms, 963–964
- release-acquire synchronization paradigm, 998, 1000–1002, 1005
- release-consume synchronization paradigm, 998–999, 1002–1003, 1005
- reopening inline namespaces, 1061–1062
- reordering data members, 178n10
- `reportError` function, 15
- reporting contracts, 1120
- representation, 480, 570
- requires clause in C++20, 486n31
- reserved identifiers, 840
- Resource Acquisition is Initialization (RAII), 388
- resource-owning objects, passing around, 771–775
- return** statements
 - disabling NRVO and implicit move, 244–246
 - moves in, 734–740
 - multiple, 1185–1187