square\_list<> design.

# Goal

Develop a C++ 11 compatible set-like templated class that implements a *square\_list*.

# Tests

|  |  |  |  |
| --- | --- | --- | --- |
| Phase | Test | Operations | Type |
| 1 | Proof of concept | square\_list()  insert()  begin()  end()  iterator operators \*,++,= | value\_type  iterator |
| 2 | Capacity methods | empty()  size()  max\_size() | size\_type |
| 3 | Forward iterator | begin()  end()  pre/post-increment  iterator ==, != |  |
| 4 | Bi-directional iterator |  | difference\_type |
| 5 | Reverse iterator | rbegin()  rend() | reverse\_iterator |
| 6 | Erase Single Element | erase() – iterator |  |
| 7 | Insert Single (Rigorous) | insert() – value |  |
| 7 | Range Modifiers | clear()  erase() – iterator  erase() – key  insert() – iterator, initializer  swap() | key\_type |
| 8 | Element access | front()  back()  find()  count()  equal\_range()  lower\_bound()  upper\_bound() | pointer |
| 9 | Constructors | copy  move  iterators  initializer\_list |  |
| 10 | Implement const[[1]](#endnote-1) | begin()  end()  cbegin()  cend()  crbegin()  crend()  at()  front()  back()  at() | const\_reference  const\_iterator  const\_reverse\_iterator |
| 11 | Relational operators | operator ==  operator !=  operator <  operator <=  operator >  operator >= |  |
| 12 | Suppress assignment (neither copy-, nor move-assignable) | operator = |  |
| 13 | Implement Allocator | ctors |  |
| 14 | Efficiency | emplace() emplace\_hint()  insert() – hint  insert() – hint, move |  |

1. Student exercise [↑](#endnote-ref-1)