

Advanced Programming in C++

Indexable Set

Design an indexable set container, `indexable_set`, with the same functionality as `std::set`, but also with the possibility to access elements by index (position). If we declare

```
indexable_set<int> ixs{ 3, 8, 2, 5, 1, 4, 7, 6, 9 };
```

`ixs` will contain nine element, ordered from 1 to 9. Index expressions as the following are to be allowed:

```
cout << ixs[0];           // 1
cout << ixs.at(8);        // 9
```

If an illegal index is given, both **`operator[]`** and `at()` shall throw `std::out_of_range`, with the message “`indexable_set::operator[] out of range`”.

```
cout << ixs[9];           // throws out_of_range
cout << ixs.at(-1);        // throws out_of_range
```

`indexable_set` shall be derived from `std::set`, be a template with the same template parameters, the same nested types, and the same operations as `std::set`.

A simple test program, `indexed-set-test.cc`, is given.

After solving this exercise (should go quickly, since only about 15 lines of effective code is required:), have a good look at the comments to the solution!