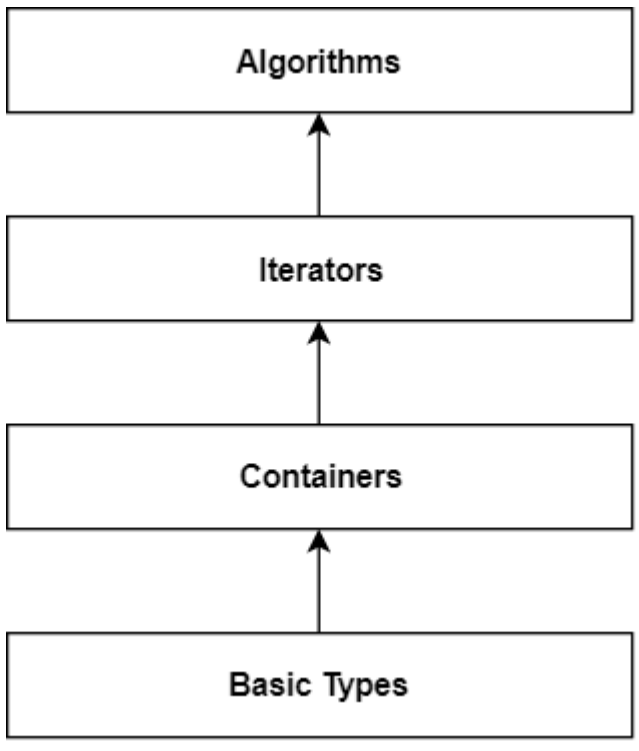
Algorithms



Container let us perform operations on basic types, regardless of the underlying data type
Iterator allow us to abstract the container being used
The STL contains pre-written algorithms that operate on iterators

Some examples

- **accumulate**

```
accumulate(InputIterator first, InputIterator last, T init)
1: get the sum of all elements in [first, last)
2: add the sum to the initial vale
```

- **nth_element**

```
void nth_element (RandomAccessIterator first, RandomAccessIterator nth, RandomAccessIterator last);
Rearranges the elements in the range [first, last) in such a way that:
the element at the n-th position is the element that would be in that position in a sorted sequence
```

- **sort and reverse**

```
void sort (RandomAccessIterator first, RandomAccessIterator last);
void reverse (BidirectionalIterator first, BidirectionalIterator last);
```

Iterator Adapaters

Ostream iterator

```
vector<int> v{3,435,1341,24,2,312}
std::copy(v.begin(), v.end(), std::ostream_iterator<int>(cout, ", "))
//This single line will print the whole vector v onto console
```

Insert Iterators

check here