Min and Max

This topic teaches the min and max algorithm by demonstrating the usage including min, max, minmax and minmax_element and explaining the algorithms

- Figure 16.13 demonstrates `min`, `max`, `minmax` and `minmax_element`.

```
cout << "The minimum of 12 and 7 is: "
     << min( 12, 7 );
cout << "\nThe maximum of 12 and 7 is: "
     << max( 12, 7 );
cout << "\nThe minimum of 'G' and 'Z' is: "
     << min( 'G', 'Z' );
cout << "\nThe maximum of 'G' and 'Z' is: "
     << max( 'G', 'Z' );
```

- Algorithms `min` and `max` determine the minimum and the maximum of two elements, respectively

C++11 now includes overloaded versions of the algorithms min and max that each receive an `initializer_list` parameter and return the smallest or largest item in the list initializer that's passed as an argument

For example, the following statement returns 7:

```
int minumum = min({10, 7, 14, 21, 17});
```

Each of these new min and max algorithms is overloaded with a version that takes as a second argument a *binary predicate function* for comparing values

```
auto result2 = minmax_element
   (items.cbegin(), items.cend());
```

- Receives two input iterators representing a range of elements
- Returns a `pair` of iterators in which `first` points to the smallest element in the range and `second` points to the largest
- A second version of this algorithm takes as a third argument a *binary predicate function* for comparing values

This topic taught the min and max algorithm by demonstrating the usage including min, max, minmax and minmax_element and explaining the algorithms