Other new features in C++ 11

# auto (type Deduction)

**Range-based for loops.**

Keep in mind these facts about range-based for:

* Automatically recognizes arrays.
* Recognizes containers that have .begin() and .end().
* Uses argument-dependent lookup begin() and end() for anything else.

int x[10] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };  
  
 // Range-based for loop to iterate through the array.  
 for( int y : x ) { // Access by value using a copy declared as a specific type.   
 // Not preferred.  
 cout << y << " ";  
 }

**Stateless lambdas**

blocks of code that begin with an empty lambda introducer [] and capture no local variables, are now implicitly convertible to function pointers.

**Auto-vectorizer.**

he compiler analyzes loops in your code and, where possible, emits instructions that use the vector registers and instructions that are present in all modern processors. (known as SSE, for*Streaming SIMD Extensions*). It is applied automatically.

**Auto-parallelizer.**

The compiler can analyze loops in your code and emit instructions that spread the calculations across multiple cores or processors. This can make the loops run faster. You must request this optimization because it is not enabled by default. In many cases, it helps to include a #pragma loop(hint\_parallel(N)) in your code immediately before the loops that you want parallelized.

**C++ Accelerated Massive Parallelism (AMP)**

**General Parallel Programming Enhancements**

* A rich task-based programming model that supports asynchrony and continuations. For more information, see [Task Parallelism (Concurrency Runtime)](http://msdn.microsoft.com/en-us/library/dd492427(v=vs.110).aspx).
* [Parallel Algorithms](http://msdn.microsoft.com/en-us/library/dd470426(v=vs.110).aspx) , which support fork-join parallelism (parallel\_for, parallel\_for with affinity, parallel\_for\_each, parallel\_sort, parallel\_reduce,parallel\_transform).
* [Concurrency-safe containers](http://msdn.microsoft.com/en-us/library/dd504906(v=vs.110).aspx), which provide thread-safe versions of std data structures such as priority\_queue, queue, vector, and map.
* The [Asynchronous Agents Library](http://msdn.microsoft.com/en-us/library/dd492627(v=vs.110).aspx), which developers can use to express dataflow pipelines that naturally decompose into concurrent units.
* A customizable scheduler and resource manager to facilitate the smooth composition of the patterns in this list. See [Task Scheduler (Concurrency Runtime)](http://msdn.microsoft.com/en-us/library/dd984036(v=vs.110).aspx).