std::async

```
template< class Function, class... Args >
std::future<typename std::result_of<Function(Args...)>::type>
async( std::launch policy, Function&& f, Args&&... args );
```

- The template function async runs the function f asynchronously (potentially in a separate thread which may be part of a thread pool) and returns a std::future that will eventually hold the result of that function call.
- Policies to spawn computation are:
  - std::launch::async
  - std::launch::deferred
  - launch::any (bitwise or async | deferred)

from http://en.cppreference.com/w/cpp

std::future

```
template< class T > class future;
template< class T > class future<T&>;
template<>          class future<void>;
```

- The class template std::future provides a mechanism to access the result of asynchronous operations.
- You can use one of these objects to get a std::future back
  - std::promise
  - std::package_task
  - std::async

from http://en.cppreference.com/w/cpp