Tasks

- A task is a unit of work (a function) which is executed asynchronously
  - Tasks are scheduled on a thread pool to optimize machine utilization

- The arguments to the task and the task results are convenient places to communicate with other tasks
  - Any function can be “packaged” into such a task

Task Systems

- Unfortunately, we don't yet have a standard async task model
  - std::async() is currently defined to be based on threads
    - This may change in C++14 and Visual C++ 2012 already implements std::async() as a task model

- Windows - Window Thread Pool and PPL
- Apple - Grand Central Dispatch (libdispatch)
  - Open sourced, runs on Linux and Android
- Intel TBB - many platform

C++14 compatible async with libdispatch

```
namespace adobe {

template <typename F, typename ...Args>
auto async(F&& f, Args&&... args)
        -> std::future<typename std::result_of<F (Args...)>::type>
{
    using result_type = typename std::result_of<F (Args...)>::type;
    using packaged_type = std::packaged_task<result_type ()>;

    auto p = new packaged_type(std::forward<F>(f), std::forward<Args>(args)...);
    auto result = p->get_future();

    dispatch_async_f(dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT, 0),
            p, [](void* f_) {
                packaged_type* f = static_cast<packaged_type*>(f_);
                (*f)();
                delete f;
            });

    return result;
}

} // namespace adobe
```