Choosing a Thread Object Solutions

Choosing a Thread Object

- What options does C++ provide for starting a task?
  - Create an std::thread object
  - Create an std::packaged_task object
  - Call std::async()
- Give the advantages and disadvantages of each one

Choosing a Thread Object

- std::async()
  - The simplest way to execute a task
  - The implementation manages the threads for us, including inter-thread communication
  - Tasks cannot be detached
  - If a task is executed with the std::launch::async option, the caller cannot leave the scope where std::async() is called until the task completes

Choosing a Thread Object

- std::packaged_task
  - The best choice if we want to represent tasks as objects
  - Gives us control over when a task runs and on which thread it runs
- std::thread
  - Can use features not supported by standard C++, by going to the underlying software thread
  - Can be detached

Recommendations

- Which situations are best suited to each of these options?
  - In the general situation, use std::async()
  - For containers of thread objects, use std::packaged_task
  - If control is needed of when tasks run, or on which thread they run, use std::packaged_task
  - For starting a detachable thread, or for more specialized requirements, use std::thread