Thread Pool Implementation
Solutions

Thread Pool Implementation

- Implement a thread pool class
- This class will manage a container of thread objects and a queue
- Users of the class will call a public member function to push tasks onto the queue
- The thread objects will pop tasks off the queue and execute them
- Write a program to exercise your class

Requirements

- The number of threads in the container must be equal to the number of processor cores provided by the hardware
- Your implementation must be thread-safe (for this exercise, assume that exceptions will be handled by the caller)
- Your implementation must be well-behaved when the queue is empty

Conclusion

- Suggest how your implementation could be improved (you are not required to write any code for this)
- As implemented in the video, all the threads will wait indefinitely, even when there will be no more work for them
  - A member function could be added which causes the threads to terminate when it is called
- As implemented in the video, there is no provision for the tasks to return a value or signal an exception
  - This could be implemented by using packaged_task instead of std::function