Locks and Mutexes in C++17 Solutions

Lock Objects

- Briefly explain what a "lock object" is in C++
  - When a lock object is created, it locks a mutex which is passed as the constructor argument
  - When the lock object goes out of scope, its destructor unlocks the mutex
  - This ensures that the mutex is always unlocked, in every possible path through the code

std::scoped_lock

- Briefly explain how std::scoped_lock differs from std::lock_guard
- Write a simple program to demonstrate the use of std::lock_guard
- Write a simple program to demonstrate the use of std::scoped_lock
  - std::scoped_lock and std::lock_guard are both lock objects
  - std::lock_guard locks a single mutex, which is passed as its constructor argument
  - std::scoped_lock locks multiple mutexes, which are passed as arguments to its constructor
  - The mutexes are locked in the order they are passed
  - They are unlocked by the destructor, in the reverse order that they were locked in