Move-only Types and RAII Solutions

Move-only types

- Give an example of a move-only class in the C++ standard library
  - std::fstream
- What problems could occur if objects of this class could be copied?
  - A copy would result in two objects owning the same file handle
  - Their writes to disk would be interleaved
  - One object could try to close the file handle while the other is using it
- When a class manages a resource using the RAII idiom, what happens when an object of that class is moved?
  - The ownership of the resource is transferred from the source object to the target object

Capture by move

- How can a lambda expression perform a capture by move?
  - We create a lambda-local variable
  - This is initialized from the variable we wish to capture, by casting the variable to an rvalue

```
fstream fs("output.txt");
[ lfs = std::move(fs) ] { ... }        // fs is moved into the lambda-local variable lfs
```

- Write a simple program which demonstrates the difference between capture by reference and capture by move