Project

You are supposed to do a programming project using (at least) one of the technologies discussed in the course. That could mean, e.g:

- Writing a Python module for performance-critical code in C++, or to aid interactive data analysis and processing of existing code in Python
- Modernize part of an existing C or C++ codebase into modern C++ (using e.g. lambdas, smart pointers, stack-based objects, range for, the auto keyword, the standard library and Boost, and the kinds of more rich abstractions and flexibility in software design that these tools allow)
- Use a parallel/asynchronous programming model with C++ abstractions, such as standard library threading, Boost MPI, Boost Asio (or even Boost Interprocess and Boost Pool)
- Use one of the scientific libraries discussed such as Trilinos, Armadillo (or the Armadillo-like Eigen)
- Analyze an existing code base using Allinea

You are supposed to either do a purely theoretical project, discussing these technologies, writing a full report. You can also do an implementation-focused project, writing a shorter technical report, in addition to any code.

Your report should contain the following:

- Scientific background – Describe your application and the computational problem
- Motivation/goal – What does the project intend to achieve
- Design choices – What technologies are you using, in what way, what other options are there
- (Implementation details – What did you discover/learn while implementing the project)
- (Results – Does it perform well? Compared to what? How did you establish this?)
- Discussion/Conclusion – Did you achieve the goal? Is your project directly applicable or more of a prototype/proof of concept? What future work do you see?

The sections in parentheses are optional, and are only required if you do an actual implementation in code. Any code you develop should also be available in a source repository.

The deadline for delivering the report is January 15. If handed in before the deadline, we will give feedback promptly.