CSE 443/543: High Performance Computing
Final Exam Study Guide

As officially scheduled by the registrar:

- **Section A**: Thursday Dec 9 from 3–5 PM in 002 Benton
- **Section B**: Tuesday Dec 7 from 3–5 PM in 002 Benton
- **Section C**: Thursday Dec 9 from 12:45–2:45 PM in 002 Benton

A different kind of finals:

As discussed, and voted on by the students during the class, we will have a different style of finals as detailed below:

- On Thu (Dec 2) class, students will provide a few short (solution should not be longer than 15-lines of C++ code) problem-solving problems that they would like to see on the final exam.
- The instructor will collect the questions at the end of this study guide.
- The final exam will be a selected subset of about 4 questions from the set of questions provided by the students.

C++ programming:

- **Basic program constructs**
  - Variables & expressions
  - if-else and if-else statements
  - switch statement
  - Looping constructs (`for`, `while`, `do-while`, range-`for`)
  - Functions/methods
    - Pass by value versus pass by reference
    - Performance and memory impact of pass-by-value
    - Using `const` keyword for parameters.
  - Default values for parameters

- **Classes and objects**
  - Using `std::string`
    - Constructors for string.
    - Methods for operating and accessing strings
    - Conversion to-and-from numeric data types to `std::string`.

- **Arrays**
  - Basics of arrays.
  - 1-D arrays
  - 2-D arrays
    - Row major organization
  - Command-line arguments

- **Vectors, iterators, and algorithms**
  - Use of vectors and iterators for processing collection of data
  - Create type aliases via the `using` clause in C++
    - Creating aliases given English description
    - Tracing aliases back to their original types.
  - Operations on a vector: adding elements, accessing elements, removing elements, etc.
  - Standard algorithms such as: `for_each`, `copy`, `copy_if`, `copy_unique`, `sort`, `find`, `min_element`, `max_element`, `min`, `max`.
  - Using external functions with algorithms
  - Using lambdas with algorithms.
  - Reading/printing/writing vectors to I/O streams

- **Hash maps (`unordered_map`)**
  - Use of `unordered_map`
  - Using `unordered_map` as associative arrays
  - Defining and using `unordered_maps` of different data types
  - Accessing values in `unordered_maps`.
    - Using `operator[]` – *i.e.*, `map[key] = value;`
    - Using `at()` method for constant maps – *i.e.*, `map.at(key)`
  - Checking values – via `map.find(key) != map.end()`
  - Iterating over all the entries in a map and processing them

```
for (auto& entry : map) {
    std::cout << entry.first << ", " << entry.second << "\n";
}
```

  - Iterating over all the entries in a map and processing them

- **Basic text file I/O operations**
  - Reading and writing data to console using `std::cin` and `std::cout`.
  - Using stream-insertion (<<) and stream-extraction (>>) operators to read and write data.
    - Understanding these operators and how they handle whitespaces.
  - Using `std::getline` method to read a full line of text
  - Using `std::ifstream` and `std::ofstream` to read/write text files.
  - Using `std::istringstream` and `std::ostringstream` to perform I/O with strings.

- **General programming concepts**
  - Fundamentals of problem solving
  - Source code, pseudo code, algorithm
  - Syntax errors and troubleshooting them
  - Semantic errors and troubleshooting them
  - Concept of data types and information that can be inferred from data types
  - Basics of files: path, absolute vs. relative path, directory vs. file. Executable vs. source file.

- **Other exercises**
  - Converting English statements to corresponding C++ statements
  - Describing C++ statements in English
  - Code walkthroughs to determine operation and output from a C++ program
  - Developing a C++ program given a functional description
  - Identifying performance or memory issues in C++ programs
  - Rewriting C++ program to address memory or performance issue

Questions supplied by the students:

1. Coming after Thu (Dec 2) class