Topics to Study for Exam 2

We encourage you to study all of the following topics for Exam 2. Feel free to ask questions if a topic is unclear. We have included the approximate importance of each category below.

Miscellaneous Material (10%)
High-level understanding of how data types are storage (e.g., require diff. amounts of storage)
Difference between pass by reference (&) and pass by value
The three different ways one might use const
Character strings
Using the String class
Using the Vector class
Recursion

1D Arrays (20%)
Understand the purpose and definition of arrays
How does one declare and/or initialize an array
Array indices start with 0
How arrays are stored in memory
Arrays are passed by 'reference' (unless const added)
Sequential search on unordered list
Selection sort on unordered list
Define base address, offset

Multi-dimensional Arrays (10%)
Understand the purpose and definition of multi-dimensional arrays
How does one declare and/or initialize a multi-dimensional array
Arrays must have bounds for all dimensions except the first
How to pass an array into a function
How to pass a row of a 2D array into a function

File Input/Output (10%)
File Streams and fstream class
5 Steps for file reading
    File reading functions: open, fail, eof, get, close, clear
5 steps for file writing
    File writing functions: open, fail, close
I/O and Classes
I/O Loop Structures (counter-controlled loops, sentinel-controlled loops, end-of-data loop)
I/O Errors

Classes (50%)

Definitions to know: OOP, class, object, composition, member functions, constructors, scope resolution operator (::), dot operator (.), unit testing, abstraction

To define a 'thing', need to define its properties and rules/behavior

How to instantiate an object

How to use dot operator to access object's properties/functions (inc. when an array of objects)

Class consists of a prototype (placed in .h) and an implementation (placed in .cpp)

Default and parameterized constructors (definition and implementation)

Difference between public and private (and purpose of private)

Purpose of accessor and mutator functions (getters and setters)

Syntax for defining a new class

Use of static const and this

Class composition and chaining

Private helper functions

Passing objects to functions