The program will likely be over arrays and vectors (the basics): searching and sorting arrays (different types). Chapters 7-13 ( we skipped 10 and 12)

**Chapter 7**

Array – allows you to store and work with multiple values of the same data type. Amount of memory used depends on the array’s data type and the number of elements. Individual elements are assigned unique subscripts which are used to access the elements. Must use a loop to display contents of array.

const int NUM\_DAYS = 6;

int days[NUM\_DAYS]; // declares an array

int days[NUM\_DAYS] = {1, 2, 3, 4, 5, 6}; //declares and initializes an array

int numbers[] = {1, 2, 3, 4, 5}; //this will create an array with 5 elements (implicit array sizing)

int numbers[7] = {1, 2, 4, 8}; //declares and partially initializes an array. Initializes the first four elements

cout << days[4]; //would print out fourth element of the days array

for (count = 0; count < NUM\_EMPLOYEES; count++){

cout << “Enter the hours worked by employee “ << (count+1) << “: “;

cin >> hours[count];} // example of loop to fill an array.

There is no bounds checking in C++.

Range based for loop – loop that iterates once for each element in an array. Each time the loop iterates it copies an element from the array to a variable.

int numbers[] = {3, 6, 9};

for (int val : numbers){

cout << val << endl;}

The above loop would display the contents of the numbers array.

Vector – a container that can store data. #include<vector>

vector<int> numbers; // no size defined

vector<int> numbers(10); size of 10 elements defined

vector<int> numbers(10, 2); in this example vector numbers has 10 elements and each element is initialized to the value 2.

**Chapter 8**

Linear search – also known as the sequential search. Uses a loop to sequentially step through an array, starting with the first element. Advantage is simplicity. Disadvantage is inefficiency.

Binary search – starts with element in the middle, repeated for the half of the array.

Bubble sort – makes passes through and compares elements of the array, certain values bubble toward the end of the array with each pass.

Select sort – moves items immediately to their final destination in the array

**Chapter 9**

& - address operator. Returns address of a variable. &amount will return address of amount var.

Pointer variable – often called pointers, designed to hold memory addresses. Allows you to indirectly manipulate data stored in other variables. You cannot multiple or divide a pointer. You can add, subtract, increment, decrement.

**Chapter 11**

Abstraction – general model of something. Dog is an abstraction.

Structure - A structure is a collection of variable types grouped together. You can refer to a structure as a single variable, and to its parts as members of that variable by using the dot (.) operator.

struct CityInfo{

string cityName;

string state;

long population;

int distance; } // initializing a structure

**Chapter 13**

Class – code that speicifies the attribultes and member functions that a particular type of object may have

Access specifiers – private and public

Accessors and Mutators – getters and setters in Java. Accessor is a member function that gets a value from a class’s member variable but does not change it (getter). Mutator is a member function that assigns a value to a member variable (setter).

Constructor – member function that is automatically called when a class object is created.

Default contructor – takes no arguments

Destructor – member function that is automatically called when an object is destroyed.

Private member function – can only be called from a function that is a member of the same class.