MAJOR TOPICS
1. Arrays
2. Multi-dimensional Arrays
3. Algorithms involving Arrays
4. Classes
5. Constructors
6. const
7. File I/O and Streams

Arrays
1. How are they declared?
2. How are they initialized at declaration?
3. What is the range for indices in an array of length N?
4. Why use a defined constant instead of a literal value when declaring an array of length N?
5. How do you declare an array, all of whose values are constant?
6. How are individual elements of an array accessed, either for reading or writing?
7. What happens if an array index is outside the legitimate range when accessing an array element?

8. How do you pass an array to a function?
9. When passing an array to a function, why is the array length or some sub-range of the array passed as well?
10. As you look at the answer to 9. above, is the array being passed by value or by reference? Be careful, two things are going on here.

11. When using a single array that will hold different data sets during the run of a program, how big should you declare the array?
12. How can you reasonably expect to work with just a portion (sub-range) of an array; i.e., what is needed to effect this?
Multi-dimensional Arrays
1. How are they declared?
2. How do you access individual elements, either for reading or writing?
3. How does passing them to functions differ from passing one-dimensional arrays?

Algorithms & Arrays
♦ Searching
  – Linear search in a 1-dimensional array
  – Linearly searching a 2-dimensional array
♦ Sorting (1-dimensional arrays)
  – Selection sort

Algorithms & Arrays (cont.)
♦ Finding maximum/minimum values in an array;
♦ Finding index of max/min values in arrays;
♦ Reversing the contents over some range in an array.
♦ Testing whether or not some sub-range constitutes a palindrome.
Algorithms & Arrays

- For this exam, you may be asked to write code to accomplish some defined task that perhaps you haven’t seen before. (Can you assimilate the information you have to solve a new problem?)

Classes

1. What are they?
2. How do you define them?
3. Use of key words public and private.
4. Why is data kept in private?
5. What are public member functions and what is their purpose?
6. How is a public member function invoked?

Classes (cont.)

7. Where are member function prototypes placed?
8. Where are function definitions normally placed?
9. Why do we use :: to define the function? Know how this is done.
10. What are accessor and mutator functions?
11. What is data encapsulation? What other terms can be used?
12. What is the principle of least privilege? Does your book define this term?
Constructors
1. What is a constructor?
2. When is a constructor invoked?
3. When you choose to write a constructor for a class you are building (you should choose to do so), what are the rules to follow regarding name, return type, and arguments.

Constructors (cont.)
4. What term is used to describe the general C++ feature that allows you to have several constructors for a class?
5. What is a default constructor?
6. If you write a constructor that has arguments, what happens if you fail to write one with no arguments?

Use of const
1. When passing a class object to a function, we often pass by reference, and in certain instances pass as reference to a const. When and why?
2. When should a class member function be declared const? How is this done, and why is it done? What happens if it isn’t done?
FILE I/O and Streams
Know the basics for opening files and checking for successful open.
How does one check for end of file.
Know the use of <<, >>, and getline(...)

Policies for this exam.
- You may bring one 8 ½ X 11 sheet of paper with “handwritten” notes of your choosing.
- Bring one or more good pencils.
- You may NOT use books, calculators, electronic devices, etc.
- If you have a cell phone make sure it is off.
- If you wear a hat with a bill on it, the bill must be turned to the rear.