1. True/False

T/F: When using delete on a pointer, that pointer is unlikely to contain the value NULL (or something equivalent).

Answer: True. The operator delete only deallocates memory pointed to by a pointer; it does not modify the value of the pointer, which is why cleaning up dangling pointers after using delete is good practice. (Page 145)

1. Multiple Choice

Which of the following is a term that refers to the generally desired method of copying classes (especially ones that use linked data), particularly in a copy constructor or when using the assignment operator? (Hint: both of these methods perform a shallow copy by default)

1. Complete copying
2. Deep copying
3. Thorough cloning
4. Shallow copying

Answer: B.

A: This is made up term. (Page N/A)

[B]: Deep copying is generally preferred because it actually allocates new, separate memory from the object being copied, then makes copies of the data in the new memory location. This makes the two objects fully separate afterwards, allowing more intuitive use of said objects. (Page 147)

C: This is a made up term. (Page N/A)

D: This is the method used by default by the compiler when the programmer does not define new behavior for the copy constructor or the assignment operator. This is generally bad because it does not create a second object, but rather doubles the number of pointers to the same memory. This often creates problems because then changing one object changes them both, which usually is undesirable. (Page 146)

1. Fill in the Blank

A class should have one (and only one) of these functions defined: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Answer: The intended answer is destructor, although other acceptable answers may exist, depending on the leniency of the grader. (Page 146)

1. Short Answer or Code

Briefly describe two advantages of array based implementations of data containers over linked ones and two disadvantages.

Answer: (Page 153-4)

Advantages: Array based implementations require less overhead (memory) than linked ones. This may be a desirable advantage depending on the intended use (platform, other program memory requirements) of the program.

Array based implementations allow for direct access to their data elements, meaning that accessing the *i*th element takes as much time to find and access as the 0 element using an appropriate pointer offset, while in a linked structure, a trail of nodes must be followed until the desired element is found. Array implementations are faster in this regard.

Disadvantages: It is difficult to appropriately allocate memory for data in array based implementations without wasting memory or using too little and having to resize too often.

Data in array based implementations requires more effort to manipulate. For example, the removal of a single data item might require that all other data items in the structure be copied over to new indices in the array, while in a linked structure, the removal of an item only requires the relinking of the nodes adjacent to the removed one.

Linked data structures are easily re-sized, by one data item at a time in fact, while array based implementations require the allocation of excessive memory and then the copying of every data item into the new memory location every time the container is re-sized.

Surely there are other comparisons to be made or other ways of stating these examples.