Chapter 18 Review

Ulysses Carlos

January 18, 2020

1 Questions

1.1 What does “Caveat emptor!” mean?

1. It means "Let the buyer beware!", Meaning that a buyer has to be responsible for being vigilant before a purchase. I’m guessing that in this case, the programmer has to be vigilant when dealing with (implementing classes, using C-style data structures (Arrays), pointers, etc.)

1.2 What is the default meaning of copying for class objects?

1. Usually, it would create a pointer that refers to the object, so like if no copy constructor/assigner was defined, in this case

   ```
   Type_T a = b;
   ```

   a points to b.

1.3 When is the default meaning of copying of class objects appropriate? When is it inappropriate?

1. Usually, the default meaning (assigning a pointer/reference to the object) is appropriate when the class object does not need to have a

distinct copy of the data. (For example, if you are copying a pointer). It is NOT appropriate if the class object needs a distinct copy of another class object and if this new class object must not be reliant on data in the other class object.

1.4 What is a copy constructor?

1. A Copy constructor is a constructor that takes a object of the same class type as an argument and allows a distinct copy of that object's data member to be stored in the new class object. For example suppose we have a class type T, with objects a and b. The constructor usually looks like this:

   ```
   T (T &object) { // Copy details here }
   ```

   You should make sure to properly allocate and deallocate memory so that no memory leaks occur.

1.5 What is a copy assignment?

1. A copy assignment is a defined operation to allow the contents of a class object to be copied to another class object, allowing each object to have distinct copies of the data. You have to define a operator= in order to do so and will usually look like this:

   ```
   Type_T& operator=(Type_T f){ // Copy details here}
   ```

   Like in the copy constructor, you must make the usage of using new and delete so that you prevent any memory leaks.

1.6 What is the difference between copy assignment and copy initialization?

1. Copy assignment allows copying of any class object past object creation, while copy initialization allows you to copy a already existing class object to a new class object through a copy constructor.

1.7 What is shallow copy? What is deep copy?

1. Shallow copy is the default method of copying in which a pointer/reference is created when a class object is assigned to a another. Both objects point/refer to the same address that holds data. Any operation done on one object will reflect on the other class object.

2. In Deep Copy, both class objects hold a distinct copy of the data, so that any change made in one object does not affect the original class object.

1.8 How does the copy of a vector compare to its source?

1. A copy of a vector(At least the standard version of a vector) has a distinct copy of the source(A distinct copy). They have the same number of objects/values, and an operator made on the copy does not affect the source.

1.9 What are the five "essential operations" for a class?

1. In the book, it mentions seven, so I'll list them out:

   - Constructors with one or more arguments
   - A default Constructor
   - A Copy Constructor to copy an object of the same type
   - A Copy Assigner to copy an object of the same type (operator=)
   - A Move Constructor to move the elements of an object of the same type
   - A Move Assigner to move the elements of an object of the same type
   - A Destructor to deallocate memory/resources taken up by the class object

   So, if you need a destructor to deallocate memory/resources, it's a good idea to define copy and move constructors and assigners.

1.10 What is an explicit constructor? Where would you prefer one over the (default) alternative?

1. An explicit constructor only allows the constructor to be called when it is (usually I think) explicitly called (i.e Type_T a (value) instead of Type_T a = (value) to prevent the constructor from being invoked in value to class conversion). I think a example would make better since here: If a constructor is not defined as explicit, a situation like this is valid:

   ```
   // Create 10 objects instead of setting v to 10
   Vector v = 10;
   void function_f(Vector & v);
   function_f(10); // Creates a vector with 10 objects.
   ```

   So it implicitly allows operators like that if the constructor is not defined as explicit.

1.11 What operations may be invoked implicitly for a class object?

1. Move operations/assignments where the compiler will implicitly use them when a object assignment occurs during a called function i.e:

   ```
   Type_T a  = function();
   ```

   where the object created in function() is destroyed at the end of the function's scope so the compiler? will call move the object members to the object a.

1.12 What is an array?

1. An array is just a chunk of memory allocated for a specific data type that usually (unless using reallloc or something) has a set size.

1.13 How do you copy an array?

1. (a) You can either use a for loop to copy an array:

   ```
   for (int i = 0; i < list_len; i++) { Copy }
   ```

   (b) Use functions like memcpy or memmove (C style, for working with code that is basically C)

   (c) Use functions like std::copy (For C++)

1.14 How do you initialize an array?

1. You can initialize an array by the following:

   ```
   Data_type array[NUM];
   Data_type array[] = {[INSERT VALUES HERE]};
   ```

   Or just declare the array and then set the values through the index.

1.15 When should you prefer a pointer argument over a reference argument? Why?

1. Because references are immutable (You can't change what it points to) You won't be able to use a reference to traverse through a C style string or through a C-style array (With a variable specifying its length of course). This includes cases where you use a Char * pointer to deal with a C-string or if you use a Type_T * pointer to traverse an array of type Type_T.

1.16 What is a C-style string?

1. A C-style string is a string that is ended with a null terminator (**0**)

1.17 What is a palindrome?

1. A palindrome is a phrase or sentence where the first half of the phrase/sentence is mirrored on the second half (Such as **radar** or the number **1001**)