Ryan Draper

Reading 5

Feb 19, 2013

1 .a) I wouldn’t have assumed that python had static length strings, but I assume it makes more as some of its classes are designed in C.

b) I also didn’t know that the regular expressions for strings, which I’ve spent some time with went back to UNIX. It did not surprise me but I do know that some of the older languages are limited to how they use and can use regular expressions.

2. a) It didn’t explain why a Boolean type could be of type ordinal which seemed logical but not clear. I just looked into it more I was confused only because of how quickly it was mentioned in the book.

b) on 246 I didn’t understand what it meant that int and some others were unsigned. Especially since C++ has a unsigned type which free more bits. I’m a little surprised that these are considered as such.

3. **static array -** Is just as it sounds the ranges are bound for the entire length of the program as well the memory allocation will stay in the same spot. This is because the binding to storage is done before run time.

**fixed stack-dynamic array –** The subscript ranges are static just as the static array the space allocation of the array is more efficient in that more than one sub-program as long as they aren’t being run in synch can use the same space for an array. Allocation and Deallocation are costly on time though due to indexing.

**stack-dynamic array –** Although once set they are the same for the life of the variable both subscript range and allocation time are done after run time. The size of the array is decided upon use and thus adds flexibility.

**fixed heap-dynamic array –** subscript ranges and storage allocation are fixed once chosen. The flexibility comes from the chose ranges are not necessary until the array is needed the costly time is allocation in the heap which takes more time than the stack.

**heap-dynamic array –** This is where all subscript ranges binding to storage and allocation are all dynamic in the program to the heap. It is the heap that is costly as these array’s can change many times over the course of the program.

4.

**static array C++**

**int numberArray[3];**

**fixed heap-dynamic C++**

**int newArray\*;**

**newArray = new (no throw) int [dynamicSize];**

**fixed stack-dyanmic C++**

**int stuff[];**

**for (int i = 0; I < 10; i++)**

**{**

**stuff[I] = (i+1) \*2;**

**}**