Unit 9 – Name Control

Section 1 - Static Elements from C

1. List two meanings for the keyword static.

The underlying concept with all uses of **static** seems to be “something that holds its position”

* Allocated once at a fixed address; that is, the object is created in a special *static data area* rather than on the stack each time a function is called. This is the concept of *static storage.*
* 2. Local to a particular translation unit (and local to a class scope in C++, as you will see later). Here, **static** controls the *visibility* of a name, so that name cannot be seen outside the translation unit or class. This also describes the concept of *linkage*, which determines what names the linker will see.

1. Describe the use of the static keyword for variables inside a function.
2. the compiler allocates storage for that variable each time the function is called by moving the stack pointer down an appropriate amount. If there is an initializer for the variable, the initialization is performed each time that sequence point is passed.

Sometimes, however, you want to retain a value between function calls.

**static** object inside a function; the storage for this object is not on the stack but instead in the program’s static data area.

This object is initialized only once, the first time the function is called,

 returns the next character in the array each time the function is called:

1. Describe the use of the static keyword for class objects inside a function.

assignment to zero has meaning only for built-in types; user-defined types must be initialized with constructor calls.

Thus, if you don’t specify constructor arguments when you define the static object, the class must have a default constructor. For example,

See example StaticObjectInFunction

1. Define the term static object destructor.

s (that is, all objects with static storage, not just local static objects as in the example above)

**exit ()** inside a destructor because you can end up with infinite recursion.

1. List two additional storage class specifiers.