First-Class Values and Umlimited Extent

First-class: A value that can be passed as a parameter, returned from a subroutine, or assigned into a variable.

Second-class: A value that can be passed as a parameter, but not returned from a subroutine or assigned into a variable.

Third-class: A value that cannot be passed as a parameter.

Unlimited extent: Objects lifetimes continues indefinitely.

Limited extent: Objects are destroyed at the end of their scope's execution.

Simple types such as integers and characters are first-class values in most programming languages. If local objects were destroyed at the end of each scope's execution, then the referencing environment captured in a long-lived closure might become full of dangling references. To avoid this problem, most functional languages specify that local objects have *unlimited extent*. Their space can only be reclaimed when the garbage collection system is able to prove that they will never be used again. Local objects in most imperative languages have *limited extent*. Space for local objects with limited extent must generally be allocated on the stack. Space for local objects with unlimited extent must generally be allocated on a heap.