

Homework 10 - Problem 1

Nicholas LaRosa

April 17, 2013

Question 1:

In Perl, an object is defined by a package (having its own namespace), and is a reference to a datatype with its own namespace. The namespace (ie. package) contains variables and subroutines, which act as the object's data members and member functions. The `bless` function creates a class reference out of a reference, therefore making an instance of the class (or object) out of a reference to some data. Therefore, the `bless` function can change a hash reference (for examples) into an object reference.

Question 2:

Hashes and arrays are both data structures available within the Perl language. With an array, elements are stored by index and thus can be accessed only through their respective index number. In other words, all elements of the array are stored one after the other in memory, so the element position is known only if the respective element's index is known. On the other hand, hashes include elements made up of keys and values. Each element of a hash has a key and value associated with it. Being stored in memory position based on its key, a key's value can quickly be retrieved from memory if the key is known. Thus, both hashes and arrays store a collection of elements, but arrays store these elements based on index (ie. storing elements one after the other in memory), while hashes store these elements based on key value (ie. storing elements in a memory address dependent upon the key value).

Question 3:

A package is used in Perl to define a new namespace or to reference an element of an existent namespace. When the current package is changed via the `package` function, the variables and subroutines declared afterwards (and before any other changing of packages) are part of this package (namespace) and thus cannot be refer-

enced or called outside of this package, unless the scope-resolution operator `::` is used. The default package or namespace is `main`, so the Perl code returns to normal functionality when in the main namespace. The `my` keyword for variable declaration allows us to limit the scope of a variable to the subroutine of which it is a member. This, way, variables cannot be modified outside of their scopes, or rather, outside of their subroutines.

Question 4:

The default variable and default array are both defined according to their context. The default variable, with symbol `$_`, is defined when looping through arrays or files with a `for` or `while` loop. When looping through an array, `$_` will equal the current element, while when looping through a filehandle, `$_` will equal the current line. The default variable will also be the default searched string, so a pattern is automatically matched to `$_` if no user-defined string is specified with the matching operator. The default array, with symbol `@_`, contains all the arguments passed to a subroutine. Both the default variable and default array cannot be modified, and thus are used to store their data into user-declared variables. Their main purpose is to be temporary storage for the values of interest depending on their context.

Question 5:

First of all, Perl allows for limited object oriented programming, which is a feature not yet seen in Bash, Csh, or Awk. Perl contains more data structures, such as the hash, that the other scripting languages lack. Additionally, Perl's subroutines are similar to the functions of Bash, but Perl allows for better function/subroutine scope declaration, due to the fact that all variables in Bash are global. Lastly, Perl requires a system call in order to call a shell command, while all the other scripting languages can make direct shell command calls. Basically, Perl seems to be a higher level scripting language, distancing itself from the shell in exchange for increased functionality.