Programming Language Design
4.1, 4.2, 4.3, 4.5, 4.6

4.1
Perl:
source: perl.org
(a) Perl does not require variable declaration before use except for 'report formats.' Perl uses the philosophy it calls DWIM, which stands for 'do what I mean.'
(b) Perl allows operator overloading
(c) Perl arrays are statically bound
(d) On the widest level is 'package' scope, which is generally referred to as 'global' and is file-wide. Lexical scope can also be file-wide (at the maximum), and then is continuously nested within blocks, which is most often subroutines.
(e) By default, variables in perl are all global, but by using the 'my' keyword, they can be lexically typed to become local to subroutines/functions

4.2
(a) Variables declared outside of functions will be global, and functions that wish to use those variables declare them with the keyword 'extern.' The ability to share variables across files depends on the compiler. GCC has no problem with declaring global variables in one file and using them in another.
(b) To hide a global variable, you declare a variable of the same name within a new block.
(c) Global variables can be dangerous because it can be difficult to know where those variables are being modified.

4.3
A declaration tells the computer to allocate new memory space for a particular type of data. A definition fills that memory space with the actual data, and is also commonly called an initialization or an assignment.

4.5
(a) Every time that we use the variables, even after they were declared, we would have to declare their types, which would be very messy, and if we were to ever do a type conversion, there might be a conflict.
(b) Overloaded functions require the context of the variables' types to know which version of the function to run. If we had variables of two different types with the same name, overloaded functions would not know what to do.

4.6
4+6
a + b
9*9*9*9*9
int a
char b
String c
No. All lvalues can be assigned to another lvalue.