

IT 240

Shell Scripting for Administrators

Chapter 10

More Control Structures

Stan J. Senesy
IT Program/CCS
New Jersey Institute of Technology

Control Structures 2

- The unless control structure acts as a logical inverse of the if structure - it only works when the test condition is anything but true
- While else may be used with unless, it tends to be confusing

Control Structures 2

- The until loop is the functional inverse of the while loop
- It runs as long as the conditional expression is false
- As soon as the condition returns true, the loop exits

Naked Blocks!

- In previous examples, we enclosed statements after a loop or other control structures in braces { }
- The 'while' control is optional however, we can just use the braces to specify a naked block of code
- The block executes once and is used principally for scope resolution

Still More Control Structures

- If multiple 'if' statements need to be tested, use the 'elif' quantifier
- Be careful - once one conditional test returns true, the structure is finished
- It does NOT behave like a C language switch statement
- The 'else' at the end is the catchall if none of the condition tests are true

Increment/Decrement

- Perl supports the C form of increment and decrement operators:

```
$bedrock++;
```

```
$bedrock--;
```

- These operators may be used either pre or post increment, just as in C

STILL More Control Structures

- Perl also supports the C version of the 'for' loop:

```
for (init; test; increment)
```

```
{
```

```
    statements
```

```
}
```

- 'foreach' loops may be written by abbreviating with 'for'

Control Operators

- The *last* operator immediately ends execution of a loop and returns control to the calling code
- The *next* operator will jump to the inside of the bottom of the current loop block
- The redo allows you to go back to the top of the current loop block

Control Operators

- Using labels can be useful in conjunction with last and other operators:

```
LINE: while (<>) {  
    foreach (split) {  
        last LINE if /__END__/;  
    }  
}
```


Logical Operators

- Perl supports the traditional logical comparison operators:
 - `&&` - AND
 - `||` - OR
- Be aware of short-circuit evaluations!
(they may be useful)

The Ternary Operator

- The ternary operator `?:` acts as a if-then-else test in a single expression

`expression ? true_expr : false_expr;`

- What if we don't want short-circuit evaluations? Use the ternary operator!