# IT 240
# Shell Scripting for Administrators

## Chapter 3
## Controlling How Scripts Run

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

# Variables

- Referencing Variables
  - *$variable_name*
  - *${variable_name}*
  - *$echo $variable_name*

# Looping and Iteration

- The shell supports the *for* loop structure:
    - *for variable in list_of_items*
    - *do*
        - *command 1*
        - *command 2*
        - *last command*
    - *done*

# Looping Over Files

- *for filename in **
- *do*
    - *command1*
    - *command2*
    - *last_command*
- *done*

# Iterative Looping

- *for i in 1 2 3 4 5 6 7 8 9 10*
- *do*
    - *command1*
    - *command2*
- *done*

# C Style Looping

- *max = upper_limit*
- *for ((i=1; i <= max; i++))*
- *do*
  - *commands*
- *done*

# Conditional Testing

- Bash also supports conditional testing using the if statement:
  - *if (condition_command) then*
    - *command1*
    - *command2*
  - *fi*

# The *else* Statement

- *if (condition_command) then*
  - *command1*
  - *command2*
- *else*
  - *command1*
  - *command2*
- *fi*

# Truth Values

- Typically, 0 (zero) is regarded as the value for true, and 1 or any non-negative number as false
- Programs typically use the *exit* command to set a return value when they finish
- It is possible to use conditional testing to see if a program (or script) finished correctly

# Redirecting Output

- Linux uses the > sign to redirect output from the standard output to wherever you direct
- Output can be redirected to a file:
  - *ls > file.txt*
- or to the null device:
  - *ls > /dev/null*

# Repetitive else

- *else* can be used repeatedly with the elif command:
  - *if (condition_command) then*
    - *command*
  - *elif (condition_command)*
    - *command*
  - *else*
    - *command*
  - *fi*

# Testing Values

- Testing for equivalence in shell scripting is a bit different than the languages you may be used to:
  - -*eq* for equivalence
  - *-ne* for not equal
  - *-gt*/*-ge* for greater than (or equal)
  - *-lt*/*-le* for less than (or equal)
- The *test* command is used to compare values

# Testing Strings

- Just as the *test* command may be used with values, strings may also be compared:
    - *“$s1”=“$s2”* equivalence
    - *“$s1”!=“$s2”* non-equivalence
    - *$s1* tests for not-null
    - *$s1 -z* zero length
    - *$s1 -n* non-zero length