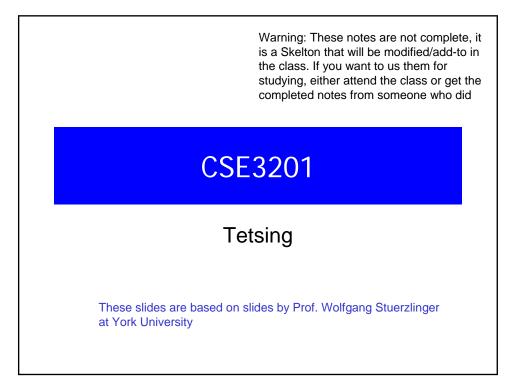
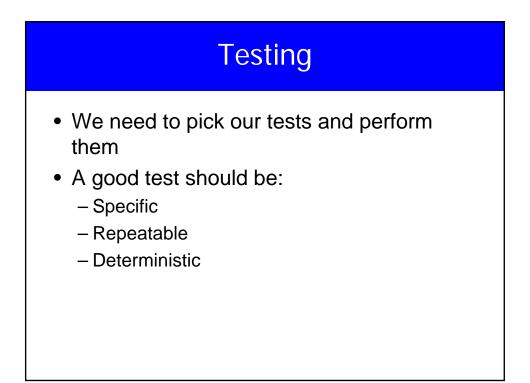
# Testing We talked about testing earlier in the course Reminder: Testing is about determining what is broken Testing is never complete Approaches to testing Black box vs. glass box



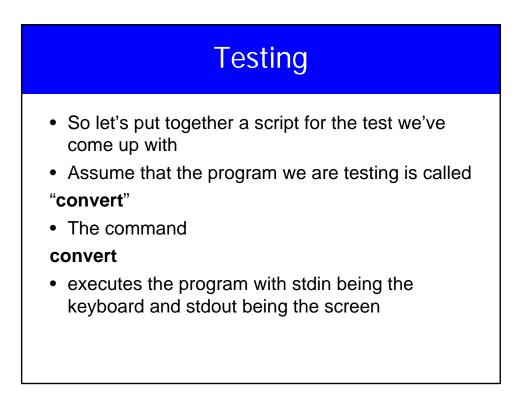
# testing and Debugging

- How programmer ensures that a program "works"
  - First creates set of test cases according to spec
  - Then creates implementation
  - Then tests implementation
  - Debugging if not all tests pass
  - Repeat until compliance with specification
- Note *all* of the above is the job of the programmer
  - Not customer, nor spec author, nor compliance tester
  - Includes creation of adequate test cases



# **Using Shell Scripts**

- Shell scripts can help with the "repeatable" and
- "deterministic" parts
- · We want to encapsulate the test in a script
- Running the script will perform the test for us and
- tell us the result
- Running the test again later should be as easy as running the script

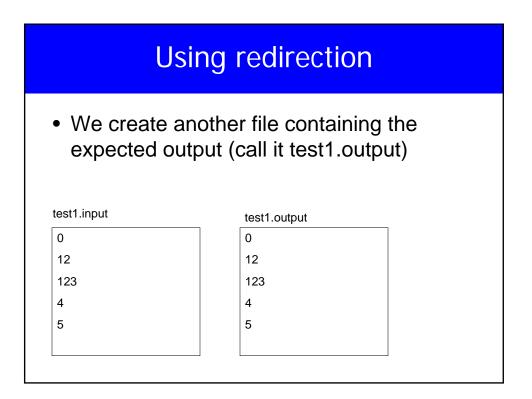


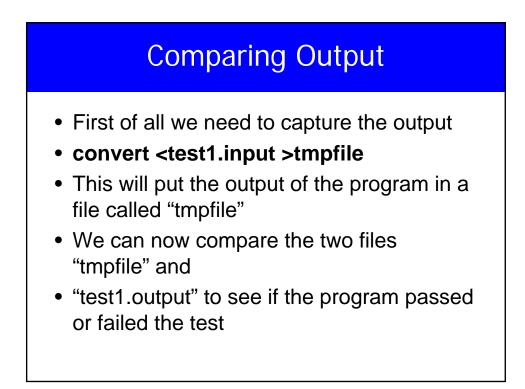
# **Using Redirection**

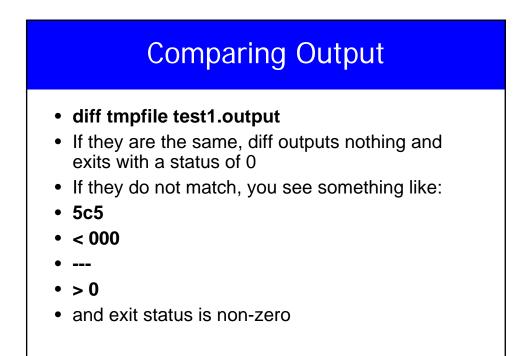
- We could perform the test ourselves by typing in the test but we want to make it easier
- Put the input in a file (called test1.input)
- Use redirection to use that file as input:
- convert <test1.input</li>

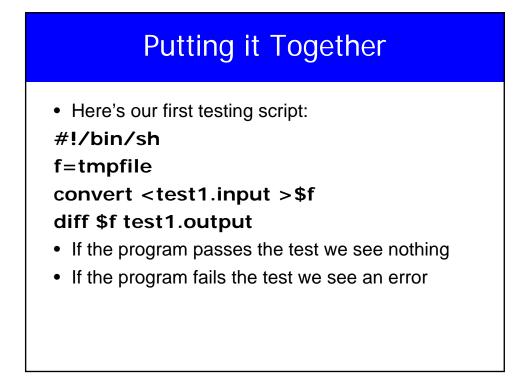
### **Using Redirection**

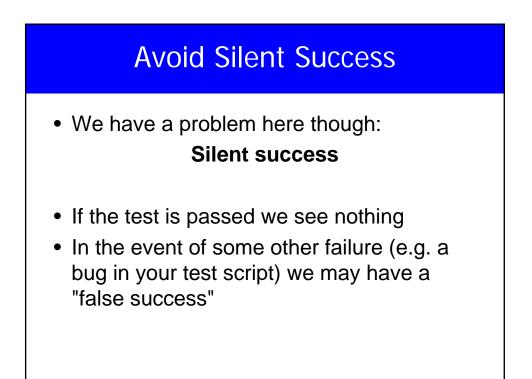
- This will run the program and give it the input and then print the output on stdout
- We could check the output by hand
   This is easy in this particular case
- However in general, it is much better to let the computer checks it.

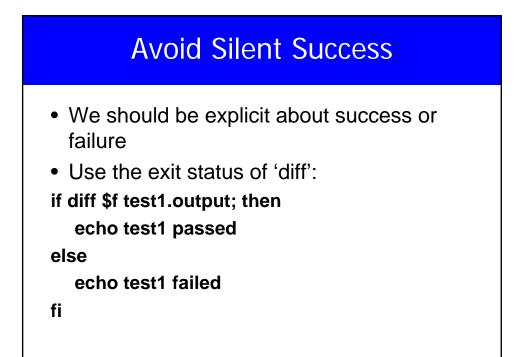












# Scripting our Script

- We've assumed that our output will be viewed by a human
- What if this is one of many tests that are being
- executed by another script?
- Should be able to signal success or failure to another program!
- · Use exit status of our script

### Scripting our Script

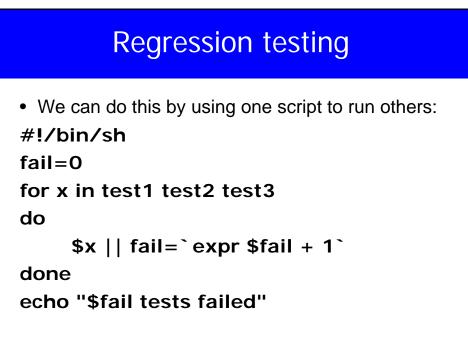
#!/bin/sh
f=tmpfile
convert <test1.input >\$f
if diff \$f test1.output; then
 echo test1 passed
 exit 0 # "true"
else
 echo test1 failed
 exit 1 # "false"
fi

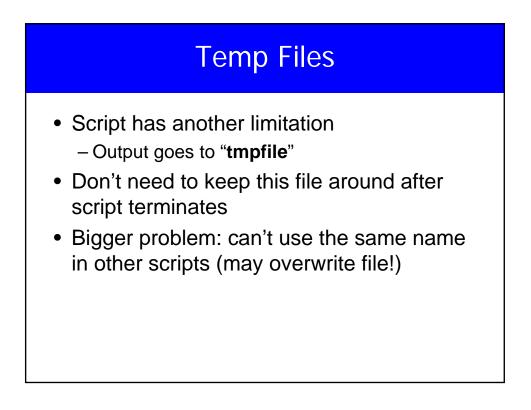
# Finally

- We can now distinguish between success and failure
- Output of our script is either
- test passed
- Or a number of lines (from diff) followed by :
- test failed
- The output of diff helps for diagnosis of failures

### **Regression testing**

- Recall: keep all of tests we come up with
- It's also a good idea to make a test for each bug you encounter while debugging
- Keep all these tests together
- After you change the program/fix a bug run all tests again





### **Temporary Files**

- Option 1: Current working directory and delete file after you are done
- Option 2: Under Unix, the path /tmp is reserved
- · for temporary files
  - Anybody can write there
  - As the name implies these files are temporary
  - On most systems, there is no guarantee the file
- will be kept around
  - Almost always automatically deleted after 1 week



- #!/bin/sh
- convert <test1.input >/tmp/tmpfile
- ...
- A little better, but we are not sure that nobody else is using the name "/tmp/tmpfile"
- A unique name would be better

## **Unique Temporary Files**

- A common approach is to use the process id of the shell (the special variable **\$\$**)
- This is an integer which is unique in the system We typically pick a name (test1) and append the
- process id to make it unique:
- /tmp/test1.\$\$
- or use the method we discussed before



- An approach which checks for conflicts:
- gettemp() {
- id=0
- while [ -f \$1.\$\$.\$id ]; do
- id=`expr \$id + 1`
- done
- echo \$1.\$\$.\$id
- }

