Controlling the Flow Through Scripts

1 Processing a List of Items (**foreach**)

```
foreach file (~cappello/i/22/C/*.c)
        chmod g+r $file
        ls -l $file
end
```

- The `break` statement will cause the shell to exit the `foreach`.

2 Performing Commands **While** a Condition Is True

```
set number = 0
while ($number < 3)
   @ number++
   echo "This is loop number $number"
   echo "Do your loop operations here"
end
```

When executed, this shell produces:

```
s> csh while
This is loop number 1
Do your loop operations here
This is loop number 2
Do your loop operations here
This is loop number 3
Do your loop operations here
```

- The `break` statement will cause the shell to exit the `while`.

3 Exiting a Command Script

- The `exit` command causes the script to be terminated, optionally setting the `status`.

  ```
  exit ( $value )
  ```

  When the parenthetical part is omitted, the status is 0.