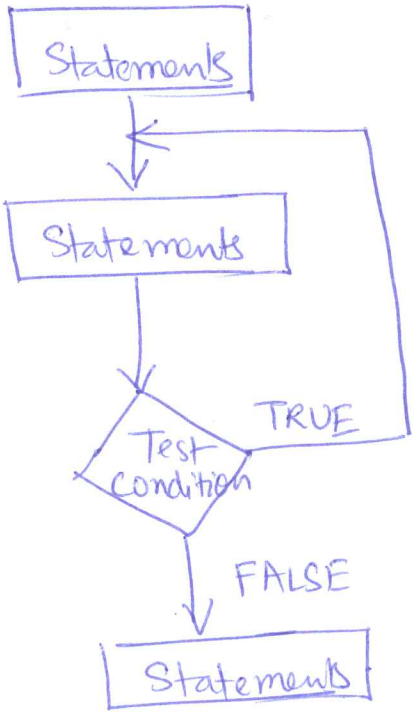
Loop Theory

- A loop is when a condition causes a specific set of programming statements to be repeated.
- The statements within the loop repeat until another condition occurs which then terminates the loop.
- Iteration or pass refers to each completion of the statements within a loop; if a loop repeats its statements 3 times, it has completed 3 iterations or passes.



(2) Infinite Loop

- A loop that theoretically repeats without end.
- It does not stop unless you terminate the script's process.
- You can terminate a script with kill signal.
- Infinite loop consume excessive amount of processing time.

Bash shell Looping commands

- for
- while
- until

(3)

CWI
Centrum Wiskunde & Informatica

for statement

- Allows you to perform a set number of iterations.

General form:

```
for variables goes from initial-value to end-value
do
    perform activity as long as variable is
    not equal to the end-value
done
```

- Two variation of the for statement
  - * Using for with a word list
  - * Using for with an arithmetic expression

for with a word list

- for statement loop through items in a word list

Syntax

```
for number in 1 2 3 4 5
do
    echo $number
done
```

for with an arithmetic expression

```
for (( num=1; n<=5; num++
do
    echo $num
done
```

num → control variable

(4)

While statement

General form:

```
while true-condition
do
    perform activity for true condition
done
```

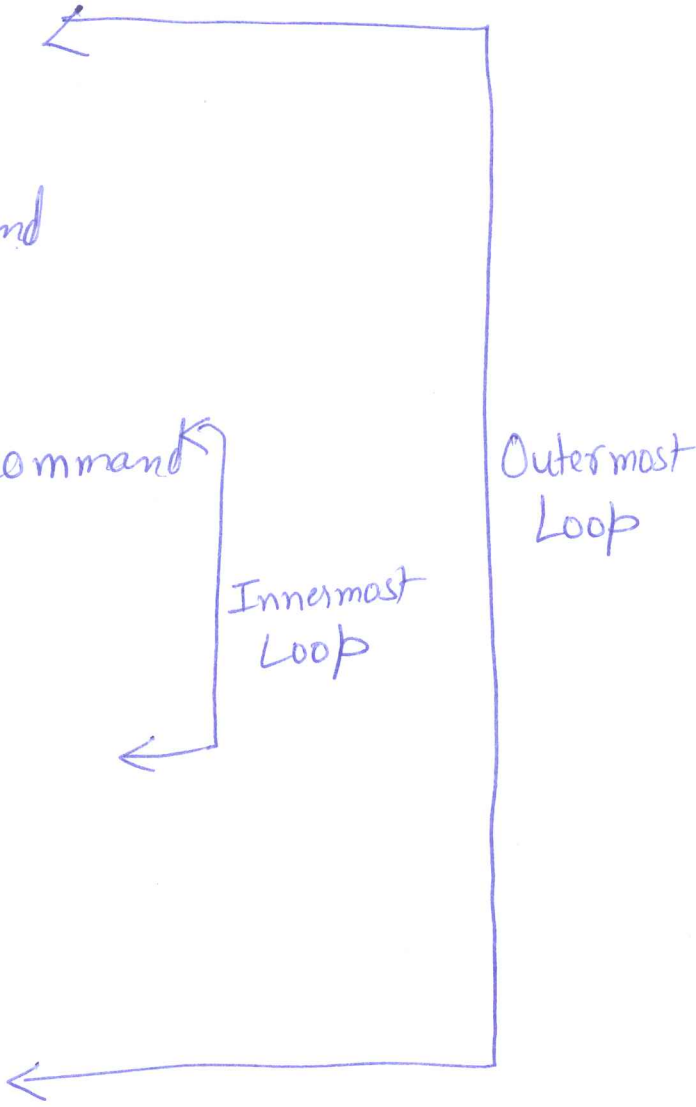
Syntax:

```
while ~~listA~~ test command
do
    Other commands
done
```

- Commands are executed in the loop as long as the exit status of test command is zero
- Exit status of test command specified must change, based on the commands run during the loop. If exit status never changes, the while loop will be stuck in an infinite loop.

Nesting loops


```
while   test command
do
    while   test command
    do
        while   test command
        do
            ____
        done
    ____
    done
____
done
```



(6) Until Statement

- Opposik logic of while
- Requires you specify a test command that normally produces a non-zero exit status.
- As long as the exit status of the test command is non-zero, the bash shell executes the commands listed in the loop.
- Once the test command returns a zero exit status, the loop stops.

```
until test command
do
    Other commands
done
```

Syntax

Controlling the Loop

break command

Simple way to escape out of a loop in progress: You can exit out of any type of loop: for, while, until.

- Use break to exit out of a single loop
- When you are working with multiple loops, the break command automatically terminates the innermost loop you are in.
- Use break n, to exit out of an outer loop from an inner loop.
  - break 2 means it will stop the next level of outer loop.
  - break 1 means exit out of current loop.

(8) SCRIPTS

FOR LOOPS

1) test1.sh → Simple script to read value from a word list

2) badtest1.sh → Problems with reading value from a word list with quotes.

3) test2.sh → How to solve the problem mentioned in previous script? Use backslash as escape character.

4) test4.sh → Reading values from a variable that contains a word list.

5) test5.sh → Reading values from a file.

6) test5b.sh → Reading values from a file when there are spaces in the word list. Use IFS.

7) test7.sh → Iterating through multiple files in a directory. Here * corresponds to all files and directories in $HOME.

ARITHMETIC EXPRESSIONS

8) test8.sh → Using multiple variables, Only one condition is possible.

(9)

CWI
Centrum Wiskunde & Informatica

9) test 10 → Example of while loop.
Example of until loop.

10) test14 → Nesting of loops.

11) test17.sh → Breaking out of a loop.
Breaking out of inner loop
Breaking out of outer loop.

12) test 23.sh → Redirecting output of a loop to a file.