CGN 2420 Programming in Mathcad

Instructor: Professor Cora Martinez, PhD
Department of Civil and Environmental Engineering
Florida International University

Objectives

- Know what Mathcad programs are and when they are useful.

- Write programs in Mathcad.

- Understand how to access program keywords through the Programming Toolbar.

- Be able to read and write programming flowcharts.

Mathcad Programs

- Essentially a multistep function.
- Very handy to write complex functions in simple form.
- Gives options for looping (for and while loops).
- Easy syntax

Basic Pieces of Program

- Function name: is used to refer where ever the program is necessary in rest of the worksheet.

- Parameters list: Is the list of all variable information which the program should know before it can do the job.

- Assignments operator : Is used to assign variable to values (locally)

Programming Toolbar

View >>Toolbars>>Programming.

Add Line [ ]]

While [CTRL +]]

Otherwise [CTRL + }]

Continue [CTRL + [ ]

Local definition [{]

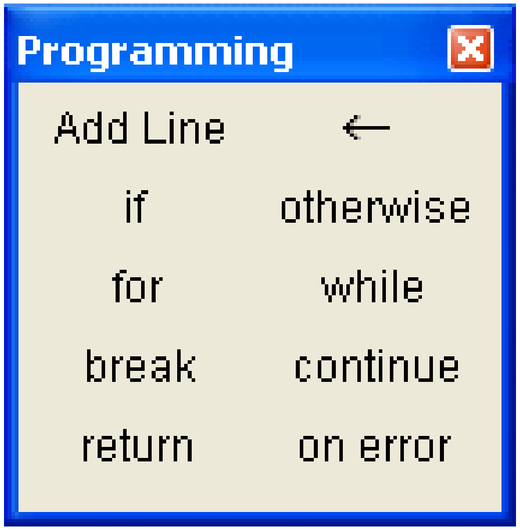
For [CTRL + “]

Return [CTRL + |]

If [{]

Break [CTRL + {]

On error [CTRL + ‘]



Program Flowchart

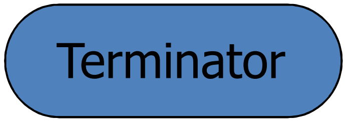
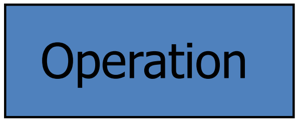
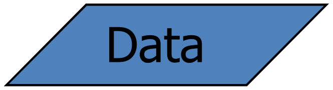
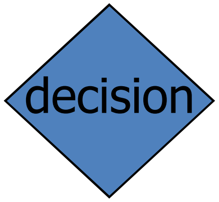
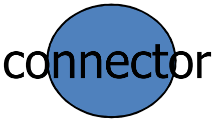
- It depicts a visual representation of a program's operation.

- It is designed to show, step by step, what the program does.

- It is used by programmers to assist in developing the program and help others understand how program works.

Program Flowchart

- Flowcharting symbols

 Terminator	Start or end of a program
 Operation	A computation step
 Data	An input or output step
 decision	A decision point in a program
 connector	Flowchart continues in another location

Program Flowchart

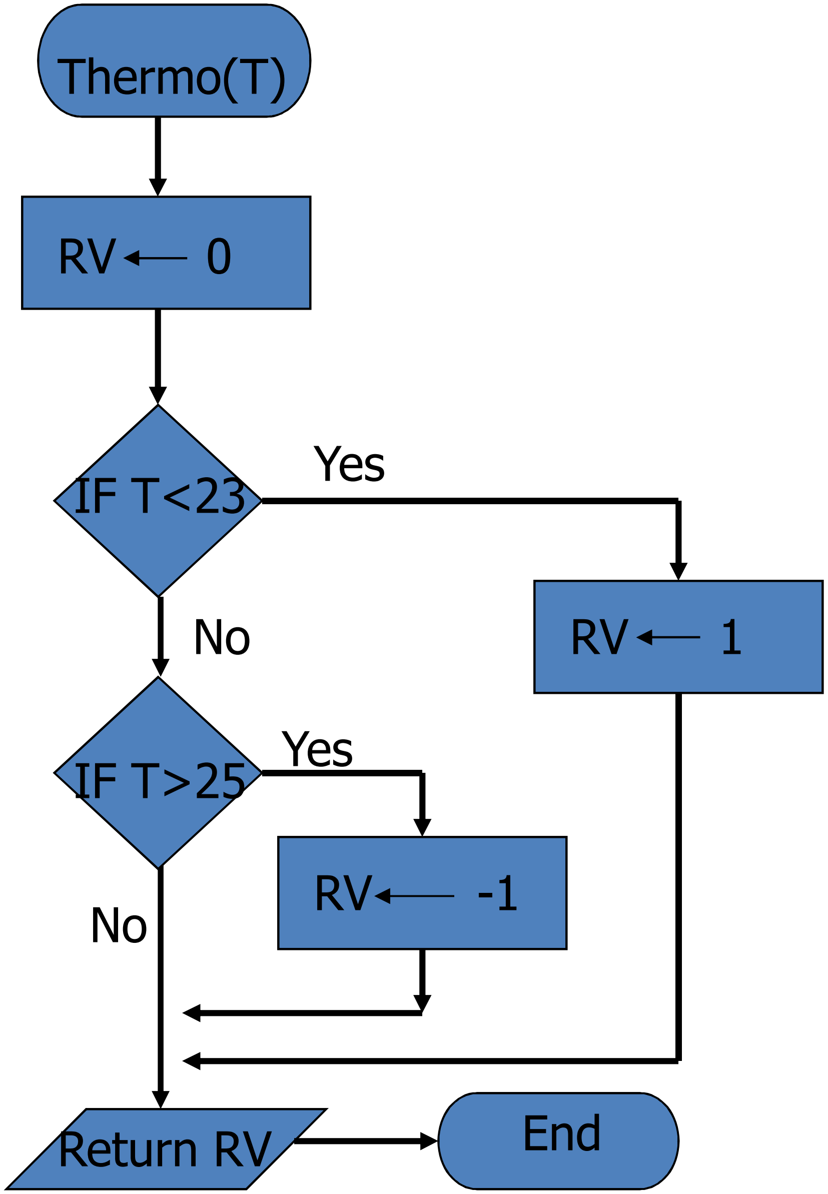
$$\text{Thermo}(T) := \left| \begin{array}{l} RV \leftarrow 0 \\ RV \leftarrow 1 \quad \text{if} \quad T < 23 \\ RV \leftarrow -1 \quad \text{if} \quad T > 25 \end{array} \right.$$

$$\text{Thermo}(20) = 1$$

$$\text{Thermo}(24) = 0$$

$$\text{Thermo}(30) = -1$$

This program check the heater temperature and states the thermostat on or off.



Basic Elements of Programming

- Data
- Input (T)
- Operations
- Output (RV)
- Conditional Execution (IF)
- Loops
- Functions

Conditional Executions

- If statement

  It is used to select from two options, depending on the result of a calculated (logical) condition.

- Otherwise statement

  This statement is used in conjunction with an If statement when the program is to do something when the condition in the If statement evaluates to false. Ex.

$$\mathrm{Thermo2}(T) := \left|\begin{array}{l} RV \leftarrow 1 \quad \text{if } \; T < 23 \\ RV \leftarrow -1 \quad \text{otherwise} \end{array}\right.$$

$$\mathrm{Thermo2}(20) = 1$$

$$\mathrm{Thermo2}(24) = -1$$

Loops

- For Loop

  Uses an iteration variable to loop a prescribed number of times. Ex.

```
addnumbers(n) :=
    $s \leftarrow 0$
    for $k \in 0..n$
        $s \leftarrow s + k$
    $s$
```

Loops

- While Loop

  Causes an action to be repeated while a condition is true

```
$\mathrm{Rootsf}(f, x) :=$
    $dx \leftarrow 1$
    $tol \leftarrow 0.0001$
    while $|dx| > tol$
        $dfxo \leftarrow \frac{d}{dx} f(x)$
        $fxo \leftarrow f(x)$
        return "error" if $dfxo = 0$
        $dx \leftarrow \frac{-fxo}{dfxo}$
        $x \leftarrow x + dx$
    $Root \leftarrow x$
```

Continue

- The **Continue** statement stops execution of the current iteration, and restarts it at the top of the nearest enclosing loop for the next iteration.

  For example, a program can add only odd integers between 0 and *n by skipping the even ones using continue.*

```
$\text{addOdds}(n) :=$
    $s \leftarrow 0$
    for $k \in 0..n$
        continue if $\text{mod}(k,2) = 0$
        $s \leftarrow s + k$
    $s$
```

Break

- The Break statement is used to stop execution of a For or While loop. It is used when some condition other than the normal loop termination is desirable to stop the loop.

  Ex:

```
demo(n) :=  $x \leftarrow n$
            while $x < 100$
                break if $x \le 1$
                $x \leftarrow x^{2.5}$
```

Return and On Error

- The **Return** operator is used in conjunction with an If statement. It stops program execution when the If statement is true, and returns the value listed.

  Ex:

  ```
  return 1.5 if  x < 100
  return 2.0 if  x > 100
  ```

- The **On Error** operator executes the right-hand argument first. If no error occurs, it returns the result of the right argument. If an error occurs, then the error is cleared and the left argument is returned.

  Ex: $F(x)= \infty$ on error $24/x$

Functions

- Functions allow a program to be broken into pieces, each one handling a single task. The programmer can then call the functions as needed to complete complex calculations.

- Functions used in programming can be Mathcad's built in functions or they can be user-written functions.