Unit 4: Control flow
Structured programming

- It is programming paradigm based on:
  - Top-down design.
  - Structure theorem
    - “Any program can be written using only three control structures: sequence, selection and iteration”

- Structured programming
  - enhances readability
  - hence maintainability
Control flow

- **Algorithm**: set of ordered steps representing the sequence of statements needed to solve a problem, without employing a specific programming language.

- **Program**: set of statements written in a language close to the machine.

- **Sequence**: a set of statements or steps executed in a specific order.

- **Flow**: the sequence of execution of an algorithm.

- **Control flow**: a mechanism allowing to change the order of execution (flow) depending on data.
Control Flow

- Linear
- Conditional
- Loops
Control Flow

Linear Execution
Linear Execution

- Execution of a set of statements one after the other, without the chance of changing the behavior of the program depending on data

- Statements
  - Assignment (containing =)
  - Expressions
  - Input / output commands
  - Calls to function or scripts
Example of Linear Execution

- Example: MATLAB program for computing the sum of two numbers

```matlab
x = input('Introduce a number:');
y = input('Introduce another number:');
sumxy = x + y;
fprintf('The sum of the numbers introduced is %d', sumxy);
```
Example of Linear Execution

- Example: MATLAB program for computing the sum of two numbers

```matlab
x = input('Introduce a number:');
y = input('Introduce another number:');
sumxy = x + y;
fprintf('The sum of the numbers introduced is %d', sumxy);
```
Control Flow

Conditional Execution
Conditional statement: if

- Syntax:

```matlab
if <expression is true>
    statement1;
    statement2;
    ...
end
```

- `<expression is true>` is a condition, an expression that Matlab evaluates to check if it is true or false

- If condition holds true then the actions are executed
Example of Conditional Statement

- Example:

```matlab
x = input('Introduce a value for x: '); 
y=4; 
if (x > y)
    disp('x is greater than y'); 
end
```
Example of Conditional Statement

Example:

Two possible execution flows

```matlab
x=input('Introduce a value for x:');
y=4;
if (x > y)
    disp('x is greater than y');
end
```
Example of Conditional Statement

- Example: The first day of each month a sentence has to be printed to state that it is the beginning of the month

```matlab
if ( dayMonth ==1)
    disp('Beginning of the month');
end
```

- Example: The first day of the month the list of tasks to be accomplished has to be printed

```matlab
if ( dayMonth ==1 )
    disp('Check Agenda');
    disp('Buy a new trasnport card');
    disp('Pay rent ');
end
```
Example of Conditional Statement

- Example: The first day of each month a sentence has to be printed to state that it is the beginning of the month

```matlab
if (dayMonth == 1)
    disp('Beginning of the month');
end
```

- Example: The first day of the month the list of tasks to be accomplished has to be printed

```matlab
if (dayMonth == 1)
    disp('Check Agenda');
    disp('Buy a new trasnport card');
    disp('Pay rent ');
end
```
When evaluating a condition the interpreter returns a result which can be either: \textit{false} (value 0) or true (any value different from 0).

The condition can be expressed in form of:

- Relational Expression
- Boolean Value
- Simple
- Compound
When evaluating a condition the interpreter returns a result which can be either: *false* *(value 0)* or *true* *(any value different from 0)*

The condition can be expressed in form of:

- Relational Expression
- Boolean Value
- Simple
- Compound

Example:
```plaintext
if (age > 18)  
  ...  
end;
```
When evaluating a condition the interpreter returns a result which can be either: *false* (value 0) or *true* (any value different from 0).

The condition can be expressed in form of:

- Relational Expression
- **Boolean Value**
- Simple
- Compound

Example:

```
if (open)
  ...
end;
```
When evaluating a condition the interpreter returns a result which can be either: \textit{false} (value 0) or \textit{true} (any value different from 0).

The condition can be expressed in form of:

- Relational Expression
- Boolean Value
- \textbf{Simple} (one condition)
- Compound

Example:

\begin{verbatim}
if (a == 18) 
  ...
end;
\end{verbatim}
When evaluating a condition the interpreter returns a result which can be either: \textit{false} (value 0) or \textit{true} (any value different from 0).

The condition can be expressed in form of:
- Relational Expression
- Boolean Value
- Simple
- \textbf{Compound} (several conditions joined using logical operators)

Example:
```
if ((a == 18) \& (amount > base *25))
    ...
end;
```
Exercise

Exercise: Write a program to help to validate the quantities introduced by the users of an ATM.

- The program asks the user to introduce a quantity (the money he/she wants to retrieve from the machine) and prints an error message when the quantity is less than 20.
Exercise

Exercise: The program asks the user to introduce a quantity and will prints an error message when the quantity is less than 20.

```matlab
quantity = input('Introduce a quantity');
if (quantity < 20)
    disp ('Error. Quantity is not valid');
end
```
Exercise

Exercise: Modify the previous program so that it also prints the error message whenever the quantity is greater than 600.
Exercise

- Exercise: Modify the previous program so that it also prints the error message whenever the quantity is greater than 600

```matlab
quantity = input('Introduce a quantity');
if (quantity < 20) | (quantity > 600)
    disp ('Error. Quantity is not valid');
end
```

The program prints the error when the quantity is smaller than 20 OR it is greater than 600. If we use AND the error would be displayed when the quantity is smaller than 20 and greater than 600.....
Exercise

- Exercise: Modify the previous program so that it also prints the error message whenever the quantity is not multiple of 10
  - Note: to calculate the remainder after a division you can use the MATLAB function `rem(num1, num2)`
Exercise

Exercise: Modify the previous program so that it also prints the error message whenever the quantity is not multiple of 10

```matlab
quantity = input('Introduce a quantity');
remainder10 = rem(quantity, 10);
if ((quantity < 20) | (quantity > 600) | (remainder10 ~= 0))
    disp ('Error. Quantity is not valid');
end
```
Exercise

- Write a program which computes the wage of an employee and prints it on screen.

The program asks the user to introduce the number of hours worked a given week and his/her hourly wage. If the employee has worked more than 40 hours during his/her work week, he/she should be paid 1.5 times his/her hourly wage for all hours worked in excess of 40.
Exercise

hours = input('Introduce the number of hours worked: ');
wage = input('Introduce your hourly wage: ');
totalWage = hours * wage;
if (hours > 40)
    extraWage = (hours - 40) * (wage / 2);
    totalWage = totalWage + extraWage;
end

fprintf('
Your payment this week is %.2f', totalWage);
Exercise

hours = input('Introduce the number of hours worked: ');
wage = input('Introduce your hourly wage: ');
if (hours <= 40)
    totalWage = hours * wage;
end
if (hours > 40)
    totalWage = 40 * wage + (hours-40) * wage * 1.5;
end
fprintf('\n Your payment this week is %.2f', totalWage);

This solution will also work. However, here we are using two different ifs for something we can do with just one. The first solution is nicer.
Conditional statement: if... else...

Syntax:
```java
if ( condition )
    trueStatement;
else
    falseStatement;
end
```

The **else** statement has no logical condition. The statements associated with it execute if the preceding if evaluates to logical 0 (false).
Example of if-else

Example

```matlab
if ( x==y )
    disp('x is equal to y');
else
    disp('x is not equal to y');
end
```
Example of if-else

- Example

```matlab
if ( x==y )
    disp('x is equal to y');
else
    disp('x is not equal to y');
end
```

Two possible execution flows
Example of if-else

Example: The player wins when the ball of the roulette stops in an even number

```matlab
numberStopped = input('Introduce the number of the roulette in which the ball stopped: '); 

typeOfNumber = rem(numberStopped, 2);
if (typeOfNumber == 0 )
    disp('Even. You win');
else
    disp('Odd. You lose');
end
```

Exercise

Write a program which asks the user to introduce his/her age, and indicates if he/she can drive or not.

Example of the program execution:

Introduce your age: 17
You are too young to drive!

Example of the program execution:

Introduce your age: 30
Ok, you can drive.
age = input('Introduce your age: ');

if (age >= 18)
    disp('OK, you can drive');
else
    disp('You are too young to drive');
end
Exercise

- Write a program which asks the user to introduce two different numbers and indicates which one is the smallest:

  Example of the program execution:
  
  Introduce one number: 5
  Introduce another number: 75
  The smallest number is 5

  Example of the program execution:
  
  Introduce one number: 30
  Introduce another number: 20
  The smallest number is 20
Exercise

num1 = input('Introduce a number: ');
num2 = input('Introduce another number: ');

if (num1 > num2)
    fprintf('The biggest number is %d', num1);
else
    fprintf('The biggest number is %d', num2);
e

Solution 1
Exercise

num1 = input('Introduce a number: ');
num2 = input('Introduce another number: ');
if (num1 > num2)
    biggest = num1;
else
    biggest = num2;
end
fprintf('The biggest number is %d', biggest);
Exercise

- The front tires of a car should both have the same pressure. Also, the rear tires of a car should both have the same pressure (but not necessarily the same pressure as the front tires). Write a program that asks the user to introduce the pressure of the four tires and writes a message that says if the inflation is OK or not.

Example of execution:

- Right front pressure: 38
- Left front pressure: 38
- Right rear pressure: 42
- Left rear pressure: 42

Inflation is OK

Example of execution:

- Right front pressure: 38
- Left front pressure: 38
- Right rear pressure: 42
- Left rear pressure: 39

Inflation is NOT OK
Exercise

RF = input('Right front pressure: ');
LF = input('Left front pressure: ');
RR = input('Right rear pressure: ');
LR = input('Left rear pressure: ');

if ((RF==LF) & (RR == LR))
    disp('Inflation is OK');
else
    disp('Inflation is NOT OK');
end
Nested conditions

- Conditional statements can be nested
- Example: Given two numbers prints on screen if the numbers are both 0, equal or not equal.
Nested conditions

- Conditional statements can be nested

- Example: Given two numbers prints on screen if the numbers are both 0, equal or not equal.

```matlab
if (x == y)
    if (x == 0)
        disp('Both are 0');
    else
        disp('x is equal to y');
    end;
else
    disp('x is not equal to y');
end
```
Conditional statement: if... elseif...

- **Syntax:**
  ```
  if ( condition1 )
    trueStatement1;
  elseif ( condition2 )
    trueStatement2;
  elseif ( condition3 )
    trueStatement3;
  ...
  else
    falseStatement;
  end;
  ```

You can use as many elseif (condition) as you want. The else statement at the end is not mandatory.
Conditional statement: if... elseif...

- Example: Given two numbers prints on screen if the numbers are both 0, equal or not equal.
Example: Given two numbers prints on screen if the numbers are both 0, equal or not equal.

Solution

```matlab
if (( x==0 ) & (y==0))
    disp('Both are 0');
elseif (x==y)
    disp('x is equal to y');
else
    disp('x is not equal to y');
end
```
Exercise

- Write a program to convert a numerical grade to a letter grade, ‘A’, ‘B’, ‘C’, ‘D’ or ‘F’, where the cutoffs for ‘A’, ‘B’, ‘C’, and ‘D’ are 90, 80, 70, and 60 respectively.

EXAMPLE:
Introduce your numberical grade: 80
Your letter grade is B
Exercise

score = input (‘Introduce your numerical grade:\’);  
if (score >=90) 
    letter = ‘A’; 
elseif (score >=80) 
    letter = ‘B’; 
elseif (score >=70) 
    letter = ‘C’; 
elseif (score >=60) 
    letter = ‘D’; 
else 
    letter = ‘F’; 
end 
fprintf(‘\n Your letter grade us %c’, letter);
Exercise

- An apple costs 0.20 euros, a coffee 1 euros, a pair of trainers 60 euros. Write a program which asks the user to introduce a quantity of euros, and prints on screen the list of names of products he/she could afford to buy.

**EXAMPLE:**
Introduce euros: 35
You can buy apples or coffees

**EXAMPLE:**
Introduce euros: 0.3
You can buy apples

**EXAMPLE:**
Introduce euros: 0.1
Sorry, not enough money …
euros = input('Introduce a euros');
if (euros < 0.20)
    disp('You can’’t buy anything...');
elseif (euros < 1)
    disp('You can buy apples');
elseif (euros < 60)
    disp('You can buy apples or coffees');
else
    disp('You can buy apples, coffee or traineers');
end
Exercise

Note:

This solution will work as well (although it is less clear)

Remember: There is always more than one way to solve the problem. Try to choose the simplest and most efficient solution

euros = input('Introduce euros');
if (euros < 0.20)
    disp('Sorry, not enough money to buy anything...');
elseif (euros >= 0.20) & (euros < 1)
    disp('You can buy apples');
elseif (euros >= 1) & (euros < 60)
    disp('You can buy apples or coffees');
elseif (euros >= 60)
    disp('You can buy apple, a coffee or traineers');
end
Exercise

Exercise: Write a program which asks the user to introduce two numbers and indicates if they are equal or which one is the smaller

EXAMPLE:
Introduce a number: 4
Introduce another number: 4
The two numbers are equal

EXAMPLE:
Introduce a number: 7
Introduce another number: 5
The smaller is 5
Exercise

numb1 = input('Introduce a number');
numb2 = input('Introduce another number');

if (numb1 > numb2)
    fprintf('The smaller is %d', numb2);
elseif (numb1 < numb2)
    fprintf('The smaller is %d', numb1);
else
    disp('The numbers are equal');
end
Exercise

Exercise: Write a program which asks the user to introduce a number between 1 and 7 and print on the screen the correspondent day of the week

EXAMPLE:
Introduce a number between 1 and 7: 5
Friday
number=input('Introduce a number');
if (number==1)
    disp('Monday');
elseif (number==2)
    disp('Tuesday');
elseif (number==3)
    disp('Wednesday');
elseif (number==4)
    disp('Thursday');
elseif (number==5)
    disp('Friday');
elseif (number==6)
    disp('Saturday');
elseif (number==7)
    disp('Sunday');
end
Conditional statement: switch

switch (variable)
  case value_1
    statement_1_1
    statement_1_2
    ...
  case value_2
    statement_2_1
    statement_2_2
    ...
  ...
  ...
  otherwise
    statement_other
end
**Conditional statement: switch**

```
switch (variable) {
    case value_1:
        statement_1_1
        statement_1_2
        ...
    case value_2:
        statement_2_1
        statement_2_2
        ...
    ...
    ...
    otherwise:
        statement_other
}
end
```

- variable containing a number, character or string
- executes if the variable is value_1
- executes if the variable is value_2
- executes if the variable does not match any other case
Example of Conditional statement Switch

switch number
  case 1
    disp('Monday');
  case 2
    disp('Tuesday');
  case 3
    disp('Wednesday');
  case 4
    disp('Thursday');
  case 5
    disp('Friday');
  case 6
    disp('Saturday');
  case 7
    disp('Sunday');
end
Example of Conditional statement Switch

THE TWO SOLUTIONS ARE VALID

switch number
    case 1
        disp('Monday');
    case 2
        disp('Tuesday');
    case 3
        disp('Wednesday');
    case 4
        disp('Thursday');
    case 5
        disp('Friday');
    case 6
        disp('Saturday');
    case 7
        disp('Sunday');
end

if (number==1)
    disp('Monday');
elseif (number==2)
    disp('Tuesday');
elseif (number==3)
    disp('Wednesday');
elseif (number==4)
    disp('Thursday');
elseif (number==5)
    disp('Friday');
elseif (number==6)
    disp('Saturday');
elseif (number==7)
    disp('Sunday');
end
Conditional statement: switch

- When the evaluated variable is
  - a number or a character the operator Matlab uses for comparing with the cases is `==`
  - a String the function Matlab uses for comparing with the cases is `strcmp`
    - `strcmp(S1,S2)` compares the strings S1 and S2 and returns logical 1 (true) if they are identical, and returns logical 0 (false) otherwise.

- The otherwise statement is optional
- If there are more than one valid ‘cases’ for the expression, only the first one will be executed
- Multiple conditions can be handled in a single case statement by enclosing the case expression in `{ }`
  - Example: case `{1, 7, 3}`

Switch only can be used to check if something is equal to…
Exercise

Exercise: Write a program which asks the user to introduce a character of the alphabet and indicates if its a vowel or not.

EXAMPLE:

Introduce a character: e
It is a vowel
Exercise

yourchar = input('Introduce a character: ','s');
switch yourchar
    case 'a'
        disp('It is a vowel');
    case 'e'
        disp('It is a vowel');
    case 'i'
        disp('It is a vowel');
    case 'o'
        disp('It is a vowel');
    case 'u'
        disp('It is a vowel');
    otherwise
        disp('It is not a vowel');
end
Exercise

```matlab
yourchar = input('Introduce a character: ','s');
switch yourchar
    case {'a', 'e', 'i', 'o', 'u'}
        disp('It is a vowel');
        disp('It is a vowel');
    otherwise
        disp('It is not a vowel');
end
```
Exercise

Exercise: Modify the weekday program so it asks the user to introduce the number of day of the week and says if it is a working day, weekend or a not correct number.

Example:
Introduce the number of day of the week: 5
It is a working day
Exercise

```matlab
weekday = input('Introduce the number of day of the week '); switch weekday
    case {1,2,3,4,5}
        disp('It is a working day');
    case {6,7}
        disp('Weekend!');
    otherwise
        disp('Sorry, incorrect number');
end
```