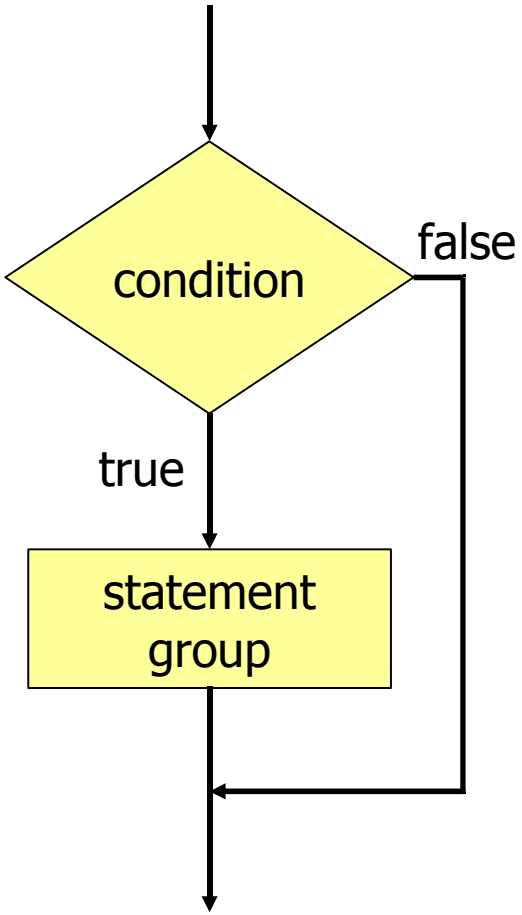
Branches

- Branches permit us to select and execute specific code sections while skipping others
- Selection of different sections depends on a condition statement
- We will learn:
  - if statement
  - switch statement

"if" Statement




```
if ( condition ),
    statement 1
    statement 2
    ...
end
```

statement group

"if" Statement

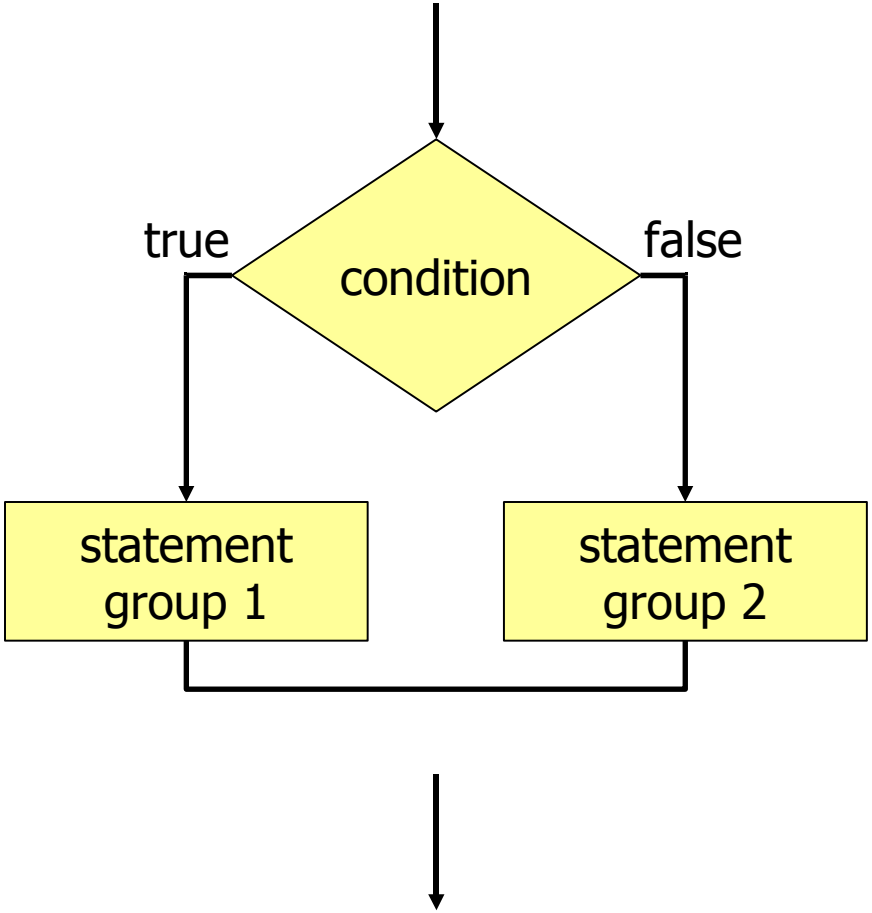
- Conditions can be:
  - any real value (0 is false, non-zero is true)
  - combination of relational and logical operators
    - e.g. (x > 0) & (x < 10)
  - logical functions
    - isempty()
    - isnumeric(), ischar()
    - isinf(), isnan()
    - exist()

Examples

- Compute the average of two exams
  - gr1 = input('Enter the grade of your 1st exam: ');
    gr2 = input('Enter the grade of your 2nd exam: ');
    if (gr1 >= 0) & (gr1 <= 100) & (gr2 >= 0) & (gr2 <= 100)
        average = (gr1 + gr2) / 2;
    end

- Display a message if the specified year is a leap year
  - year = input('Enter a year: ');
    if (mod(year,4) == 0) & ...
      (mod(year,100) ~= 0 | mod(year,400) == 0)
        disp([num2str(year) ' is a leap year'])
    end

"if-else" Statement




```
if ( condition ),
    statement 1
    statement 2
    ...
else
    statement 1
    statement 2
    ...
end
```

statement group 1

statement group 2

Example: “Leap year”

- Display a message whether or not the specified year is a leap year
  - year = input('Enter a year: ');
    if (mod(year,4) == 0) & ...
      (mod(year,100) ~= 0 | mod(year,400) == 0)
        disp([num2str(year) ' is a leap year'])
    else
        disp([num2str(year) ' is not a leap year'])
    end

Example: “Min and max”

- Compute the maximum and minimum of three numbers entered by the user
  - using *max* and *min* built-in functions:
    - ```
      n1 = input('Enter the 1st number: ');
      n2 = input('Enter the 2nd number: ');
      n3 = input('Enter the 3rd number: ');
      max_no = max([n1 n2 n3]);
      min_no = min([n1 n2 n3]);
      ```
  - without using *max* and *min* built-in functions:

?
      ```

Example: “Min and max”


```
n1 = input('Enter the 1st number: ');
n2 = input('Enter the 2nd number: ');
n3 = input('Enter the 3rd number: ');
if n1 > n2
      min_no = n2;
      max_no = n1;
else
      min_no = n1;
      max_no = n2;
end
if n3 < min_no
      min_no = n3;
end
if n3 > max_no
      max_no = n3;
end
```

Nested "if" Statement

- Two if statements can be nested
  - One of them lies entirely within a single code block of the other one

```
if (condition 1)
    …
    if (condition 2)
        …
    end
end
```

Example: "Convert case"

- Write a program that converts the lowercase letters to uppercase and uppercase letters to lowercase. This program gives a message for other characters.
  - using *lower* and *upper* built-in functions:

```
c = input('Enter a character: ','s');
if (c >= 'A') & (c <= 'Z')
    new_c = lower(c);
    disp(['The lower case of ' c ' is ' new_c]);
else
    if (c >= 'a') & (c <= 'z')
        new_c = upper(c);
        disp(['The upper case of ' c ' is ' new_c]);
    else
        disp([c ' is not a letter']);
    end
end
```

Example: "Convert case"

- Write a program that converts the lowercase letters to uppercase and uppercase letters to lowercase. This program gives a message for other characters.
  - without using *lower* and *upper* built-in functions:

```
c = input('Enter a character: ','s');
if (c >= 'A') & (c <= 'Z')
    new_c = char('a' + (c - 'A'));
    disp(['The lower case of ' c ' is ' new_c]);
else
    if (c >= 'a') & (c <= 'z')
        new_c = char('A' + (c - 'a'));
        disp(['The upper case of ' c ' is ' new_c]);
    else
        disp([c ' is not a letter']);
    end
end
```

Example: "Roots of a quadratic eqn."

- Write a program to solve for the roots of a quadratic equation
  - For the quadratic equation of the form

$$ax^2 + bx + c = 0$$

  - The solution is

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

    - If $b^2 - 4ac > 0$ then two distinct roots
    - If $b^2 - 4ac = 0$ then two identical roots
    - If $b^2 - 4ac < 0$ then no real roots

Example: "Roots of a quadratic eqn."

```
a = input('Enter a: '); b = input('Enter b: '); c = input('Enter c: ');
if a == 0
    disp('This equation is not quadratic');
else
    discriminant = b^2 - 4 * a *c;
    if (discriminant > 0)
        x1 = (-b + sqrt(discriminant)) / (2 * a);
        x2 = (-b - sqrt(discriminant)) / (2 * a);
        fprintf('This equation has two real roots: %.2f and %.2f\n',x1,x2);
    else
        if (discriminant < 0)
            fprintf('This equation has complex roots\n');
        else
            x1 = (-b + sqrt(discriminant)) / (2 * a);
            fprintf('This equation has two identical roots: %.2f\n',x1);
        end
    end
end
```

Example: "Roots of a quadratic eqn."

```
a = input('Enter a: '); b = input('Enter b: '); c = input('Enter c: ');
if a == 0
disp('This equation is not quadratic');
else
discriminant = b^2 - 4 * a *c;
if (discriminant > 0)
x1 = (-b + sqrt(discriminant)) / (2 * a);
x2 = (-b - sqrt(discriminant)) / (2 * a);
fprintf('This equation has two real roots: %.2f and %.2f\n',x1,x2);
else
if (discriminant < 0)
fprintf('This equation has complex roots\n');
else
x1 = (-b + sqrt(discriminant)) / (2 * a);
fprintf('This equation has two identical roots: %.2f\n',x1);
end
end
end
```

indentation is important for readability and understandability

Example: “Roots of a quadratic eqn.”

- How to test our program?
  - Using at least one test case for each possible path
  - Testing how your program works for possible critical values

Example: "Letter grade assignment"

- Assign the letter corresponding to a given numerical grade

  90 – 100 → A

  80 – 89 → B

  70 – 79 → C

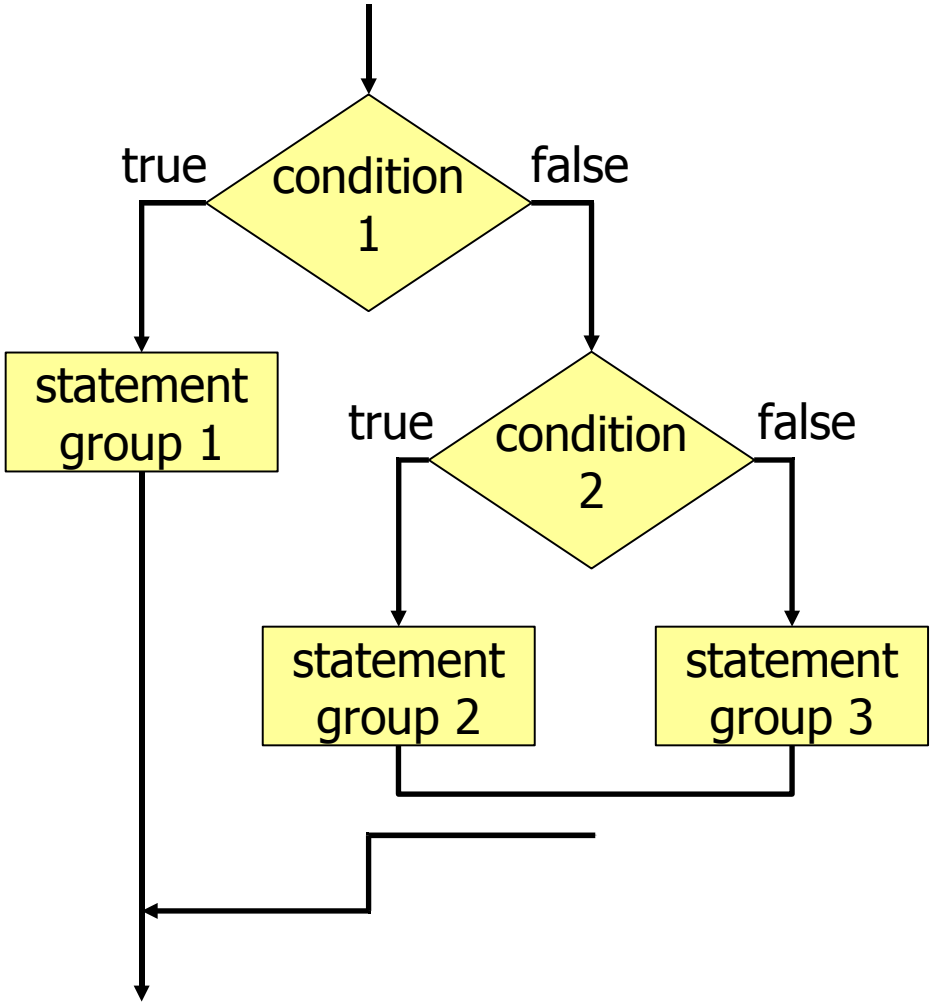
  60 – 69 → D

  0 – 59 → F

Example: “Letter grade assignment”

```
grade = input('Enter your numerical grade: ');
if grade < 0 | grade > 100
      disp('Invalid grade');
else
      if grade >= 90
            letter = 'A';
      else
            if grade >= 80
                  letter = 'B';
            else
                  if grade >= 70
                        letter = 'C';
                  else
                        if grade >= 60
                              letter = 'D';
                        else
                              letter = 'F';
                        end
                  end
            end
      end
      disp(['Your letter grade: ' letter]);
end
```

"if-elseif-else" Statement




```
if ( condition 1 ),
      statement 1
      statement 2          } statement group 1
      ...
elseif ( condition 2 ),
      statement 1
      statement 2          } statement group 2
      ...
else
      statement 1
      statement 2          } statement group n
      ...
end
```

Example: “Letter grade assignment”

```
grade = input('Enter your numerical grade: ');
if grade < 0 | grade > 100
    disp('Invalid grade');
else
    if grade >= 90
        letter = 'A';
    elseif grade >= 80
        letter = 'B';
    elseif grade >= 70
        letter = 'C';
    elseif grade >= 60
        letter = 'D';
    else
        letter = 'F';
    end
    disp(['Your letter grade: ' letter]);
end
```

"switch" Statement


```
switch ( expression ),
case value 1,
    statement 1
    statement 2
    ...
case value 2,
    statement 1
    statement 2
    ...
otherwise
    statement 1
    statement 2
    ...
end
```

← expression is a scalar or string constant

statement group 1

statement group 2

optional statement group that is executed if none of the cases is satisfied

Example: “Month names”

- Display the name of the corresponding month when a month number is given
  - 1 → January

    2 → February

    …

    12 → December

Example: “Month names”


```
switch (month_no)
      case 1,      disp('January');
      case 2,      disp('February');
      case 3,      disp('March');
      case 4,      disp('April');
      case 5,      disp('May');
      case 6,      disp('June');
      case 7,      disp('July');
      case 8,      disp('August');
      case 9,      disp('September');
      case 10,     disp('October');
      case 11,     disp('November');
      case 12,     disp('December');
      otherwise, disp('Invalid month');
end
```

Example: "Letter grade assignment"

- Assign the letter corresponding to a given numerical grade using "switch" statement

  90 – 100 → A

  80 – 89 → B

  70 – 79 → C

  60 – 69 → D

  0 – 59 → F

Example: “Letter grade assignment”

```
grade = input('Enter your numerical grade: ');
if grade < 0 | grade > 100
    disp('Invalid grade');
else
    switch (floor(grade/10))
        case 10,
            letter = 'A';
         case 9,
            letter = 'A';
        case 8,
            letter = 'B';
        case 7,
            letter = 'C';
        case 6,
            letter = 'D';
        otherwise
            letter = 'F';
    end
    disp(['Your letter grade: ' letter]);
end
```

"switch" Statement


```
switch ( expression ),
case {value set 1},
    statement 1
    statement 2
    ...
case {value set 2},
    statement 1
    statement 2
    ...
...
otherwise,
    statement 1
    statement 2
    ...
end
```

statement group 1

statement group 2

statement group n

Example: “Odd or even”

- Determining odd and even numbers in the range of 1 and 10

```
switch (value),
        case {1,3,5,7,9},
                disp('Odd number');
        case {2,4,6,8,10},
                disp('Even number');
        otherwise,
                disp('Out of range');
end
```

Example: "Number of days"

- Find a number of days in a specified month
  - Jan, Mar, May, Jul, Aug, Oct, Dec → 31

    Apr, Jun, Sep, Nov → 30

    Feb → 29 (for a leap year)

    28 (for a non-leap year)

Example: “Number of days”


```
month_name = input('Enter the month name: ','s');
switch (month_name)
      case {'Jan','Mar', 'May', 'Jul', 'Aug', 'Oct', 'Dec' },
            day_no = 31;
      case {'Apr','Jun','Sep','Nov'},
            day_no = 30;
      case {'Feb'},
            year = input('Enter the current year: ');
             if (mod(year,4) == 0) & (mod(year,100) ~= 0 | mod(year,400) == 0)
                  day_no = 29;
             else
                  day_no = 28;
             end
      otherwise,
            day_no = 0;
end
if day_no == 0,
      disp('Invalid month name');
else
      disp(['There are ' num2str(day_no) ' days in ' month_name]);
end
```

Example: "Unit converter"

- Implement a unit converter that
  - takes a value in cm and
  - converts it to one of the following units
    - mm, millimeter (1cm = 10mm)
    - cm, centimeter
    - m, meter (1m = 100cm)
    - in, inch (1in = 2.54 cm)

Example: “Unit converter”


```
x = input('Length (in cm): ');
u = input('Unit: ', 's');
switch (u),
      case {'cm','centimeter'},
            disp([num2str(x) 'cm'] );
      case {'mm','millimeter'},
            disp([num2str(10 * x) 'mm'] );
      case {'m','meter'},
            disp([num2str(x/100) 'm'] );
      case {'in','inch'},
            disp([num2str(x/2.54) 'in'] );
      otherwise,
            disp('Unknown unit');
end
```