



UNIVERSIDAD  
NACIONAL DE  
HURLINGHAM



Universidad  
Nacional  
de San Martín

# Control Flow Statements

## Loops - for, while etc

Leandro Luciano Gagliardi  
lgagliardi@unsam.edu.ar

# Tipo de declaraciones

- **Declaraciones de selección:**
  - if ... else
  - switch
- **Declaraciones iterativas:**
  - while
  - do .. while
  - for
- **Declaraciones de salto:**
  - break
  - continue
  - goto

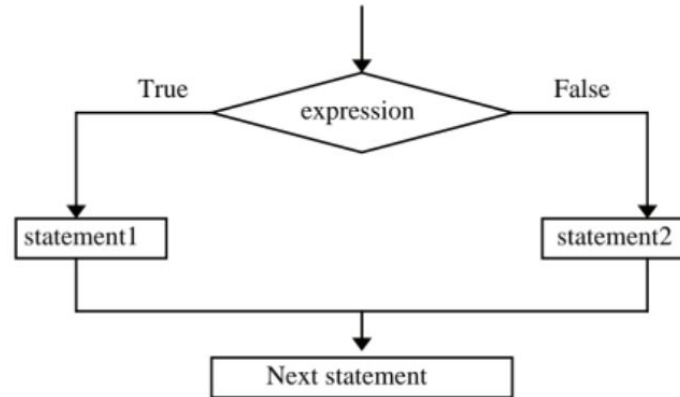
# if ... else

```
if(expresión)
    declaración 1;
else
    declaración 2;
```

```
if(expresión){
    declaración 1;
    ...
}
else
    declaración 2;
```

```
if(expresión)
    declaración 1;
else{
    declaración 2;
    ...
}
```

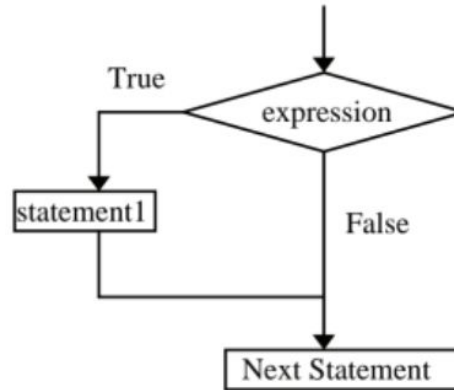
```
if(expresión){
    declaración 1;
    ...
}
else{
    declaración 2;
    ...
}
```



# if ... else

```
if(expresión)  
    declaración 1;
```

```
if(expresión){  
    declaración 1;  
    ...  
}
```



# if ... else

```
if(expresión 1){  
    declaración 1;  
    ...  
    if(expresión 2){  
        declaración 3;  
        ...  
    }  
    else{  
        declaración 4;  
        ...  
    }  
}  
else{  
    declaración 2;  
    ...  
}
```

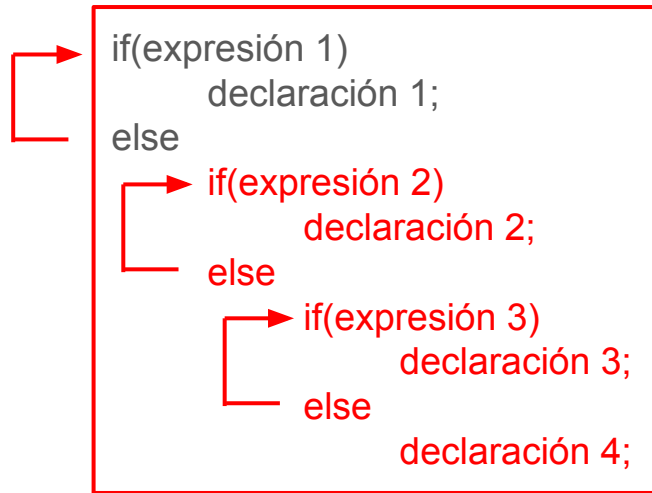
```
if(expresión 1)  
    declaración 1;  
    if(expresión 2)  
        declaración 3;  
    else  
        declaración 4;  
else  
    declaración 2;
```

# Problema del else colgado (dangling else)

```
if(expresión 1)
    if(expresión 2)
        declaración 3;
    else
        declaración 2;
```

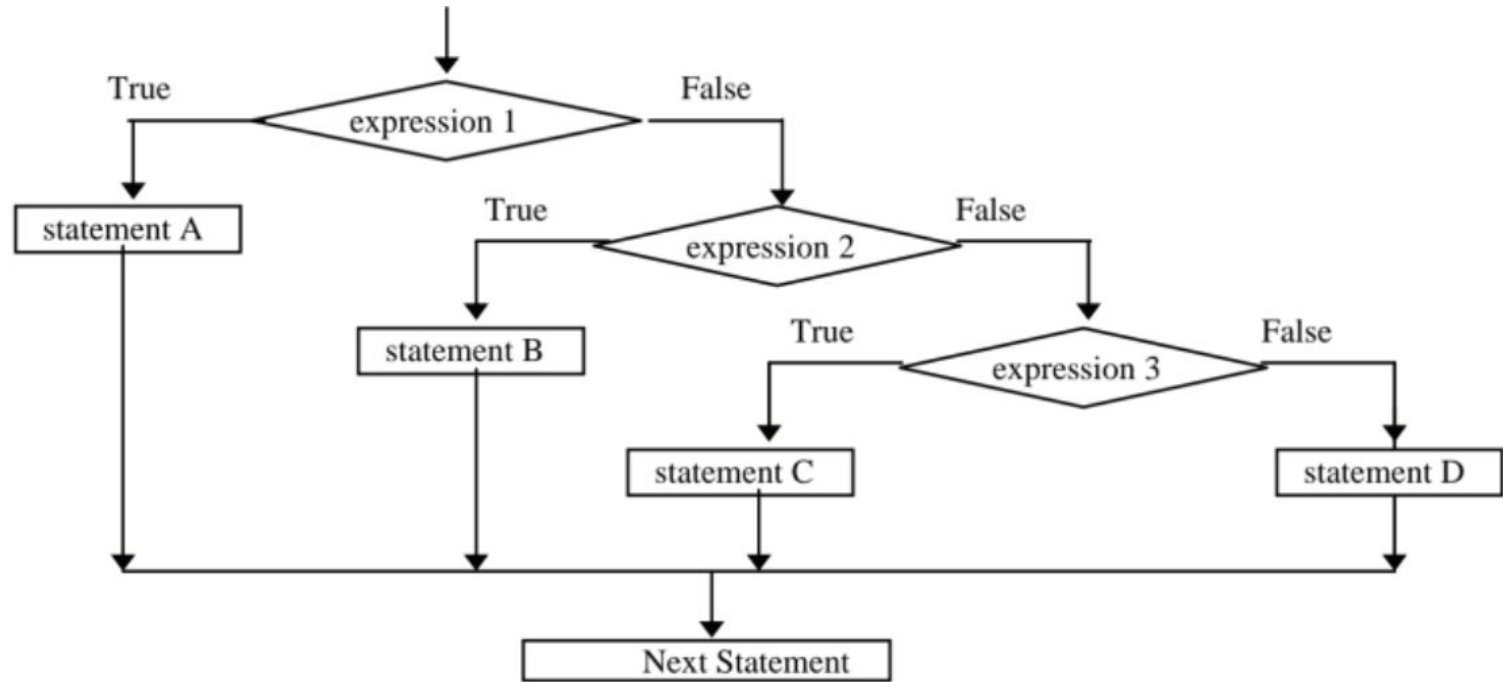
```
if(expresión 1){
    if(expresión 2)
        declaración 3;
}
else
    declaración 2;
```

# Escalera de if - else



```
if(expresión 1)
    declaración 1;
else if(expresión 2)
    declaración 2;
else if(expresión 3)
    declaración 3;
else
    declaración 4;
```

# Escalera de if - else

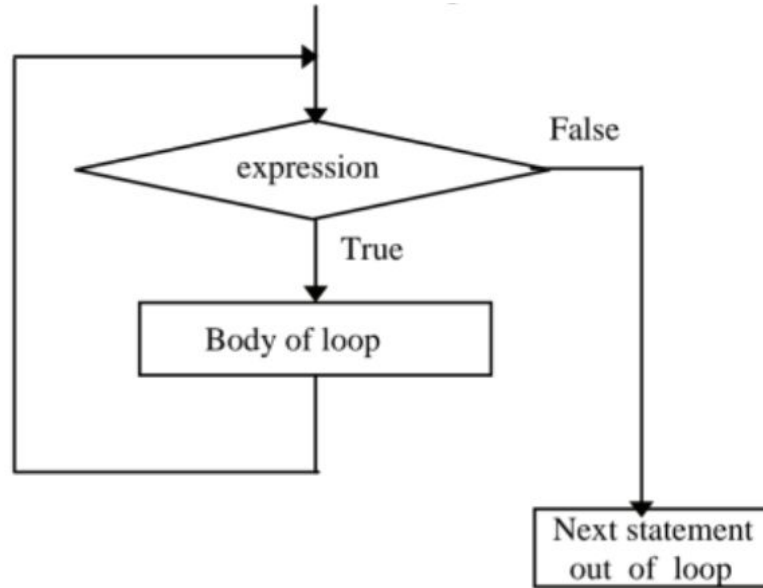




# While

```
while(expresión)  
    declaración 1;
```

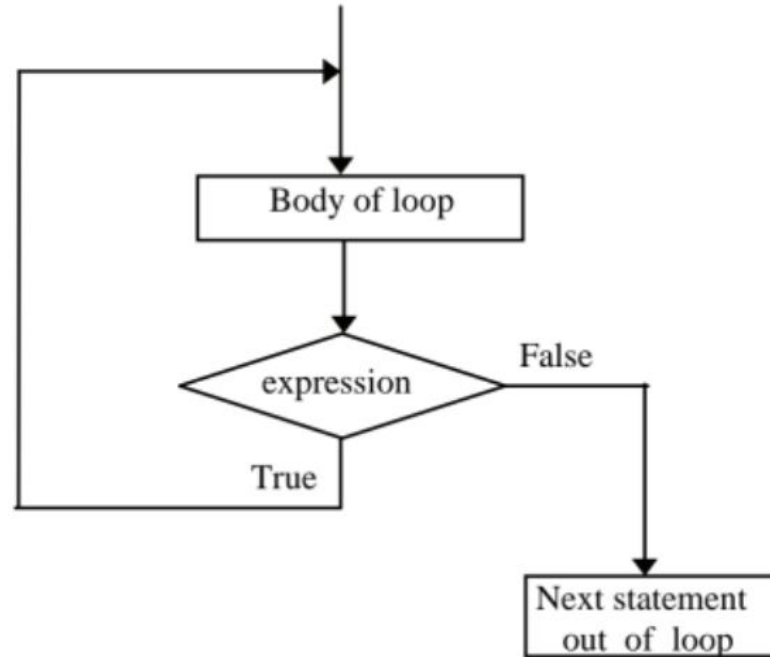
```
while(expresión){  
    declaración 1;  
    declaración 2;  
    ...  
}
```



# do ... While

```
do  
    declaración 1;  
while(expresión);
```

```
do{  
    declaración 1;  
    declaración 2;  
    ...  
}while(expresión);
```

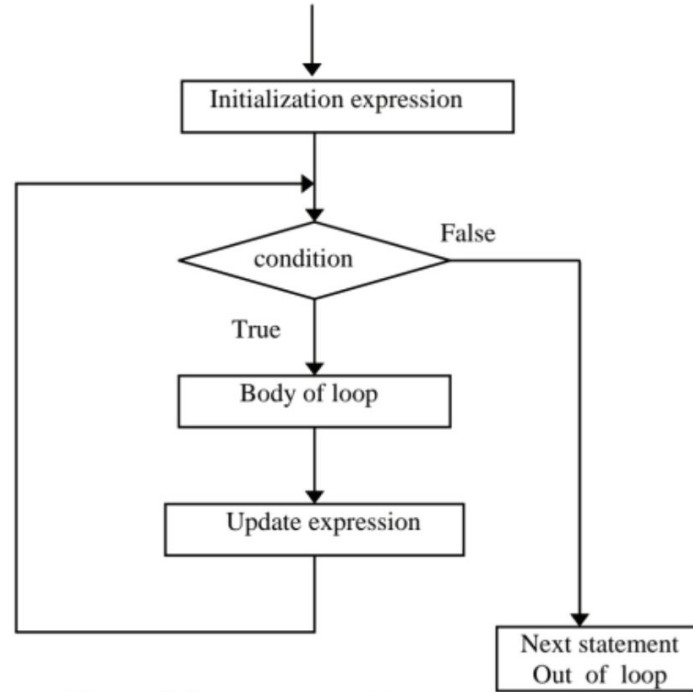


# for

```
for(expresión 1; expresión 2; expresión 3)  
  declaración 1;
```

```
for(expresión 1; expresión 2; expresión 3){  
  declaración 1;  
  declaración 2;  
}
```

```
expresión 1;  
while(expresión 2){  
  declaración 1;  
  ...  
  expresión 3;  
}
```



# loops anidados

```
for(expresión 1; expresión 2; expresión 3){  
    declaración 1;  
    for(expresión 4; expresión 5; expresión 6){  
        declaración 2;  
        for(expresión 7; expresión 8; expresión 9)  
            declaración 3;  
        declaración 4;  
    }  
    declaración 5;  
}
```

# Un pequeño desafío :)

- Diseñar un programa que permita ingresar un número entero y que sume todos los dígitos.  
Ejemplo:

>> Ingrese un número entero:

>> 1234

>> Suma de todos los dígitos =  $1 + 2 + 3 + 4 = 10$

# Una solución posible

```
#include <stdio.h>

int main(){

    int num, suma = 0;
    printf("Enter an integer number: ");
    scanf("%d", &num);
    printf("\n\n");

    while(num){
        suma += num%10;
        num = num/10;

        printf("suma = %d\n", suma);
        printf("num = %d\n\n", num);
    }
    printf("total = %d", suma);

    return 0;
}
```

Enter an integer number: 91372

suma = 2  
num = 9137

suma = 9  
num = 913

suma = 12  
num = 91

suma = 13  
num = 9

suma = 22  
num = 0

# loops infinitos

```
k = 0;  
for(int i = 0; i < 5; k++);
```

```
k = 1;  
while(k < 5){  
    ...  
}
```

```
while(k = 5){  
    ...  
}
```

```
float k = 2.0;  
while(k != 3.0){  
    k =+ 0.2;  
}
```

```
k = 1;  
while(k < 5);
```

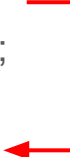
```
while(1){  
    ...  
}
```

```
do{  
    ...  
}while(1);
```

```
for( ; ; ){  
    ...  
}
```


# break

```
k = 0;
for(int i = 0; i < 5; k++){
    if(4 <= i)
        break;
    printf("%d", i);
}
...
```

A red line originates from the **break;** statement and extends to the right, then turns downwards and left, ending with an arrowhead pointing to the closing curly brace of the for loop, indicating an immediate exit from the loop.


# continue

```
k = 0;
for(int i = 0; i < 5; k++){
    if(4 <= i)
        continue;
    printf("%d", i);
}
...
```

A red line originates from the **continue;** statement and extends to the left, then turns upwards and right, ending with an arrowhead pointing to the opening curly brace of the for loop, indicating that the rest of the current iteration is skipped and the next iteration begins.



# goto

```
goto label;  
...  
declaración 1;  
...  
label:   
declaración 2;  
...
```

# switch

```
switch (expresión){  
    case constante 1:  
        declaración 1;  
        ...  
    break;  
    case constante 2:  
        declaración 2;  
        ...  
    break;  
    ...  
    case constante N:  
        declaración N;  
        ...  
    break;  
    default:  
        declaración N + 1;  
        ...  
    break;  
}
```



UNIVERSIDAD  
NACIONAL DE  
HURLINGHAM



Universidad  
Nacional  
de San Martín

# Extra Slides

Leandro Gagliardi  
[lgagliardi@unsam.edu.ar](mailto:lgagliardi@unsam.edu.ar)