Semana-06

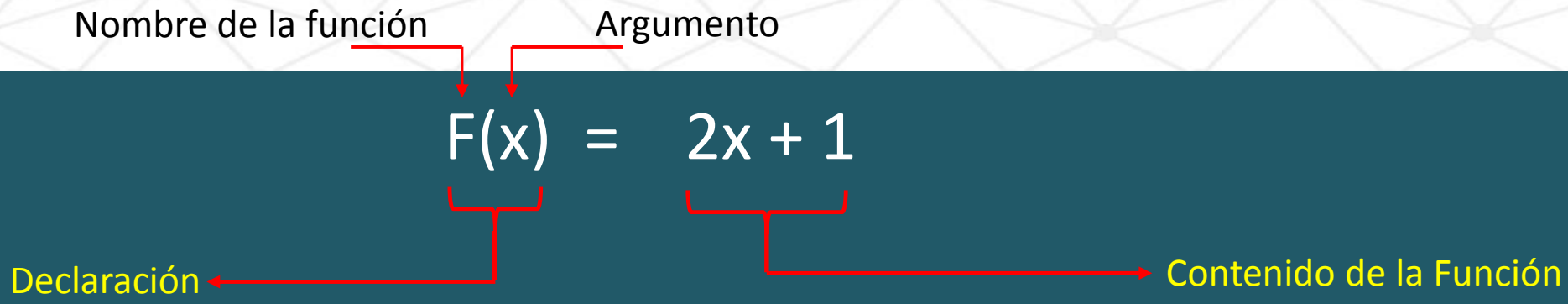
IMPLEMENTACIÓN DE FUNCIONES

Temario

- Implementación de funciones con retorno y sin retorno en C++

FUNCIONES - INTRODUCCIÓN

En matemática: **Ejemplo 1**



Aplicación de la función:

21 ← F(10)

6 ← 3 + F(1)

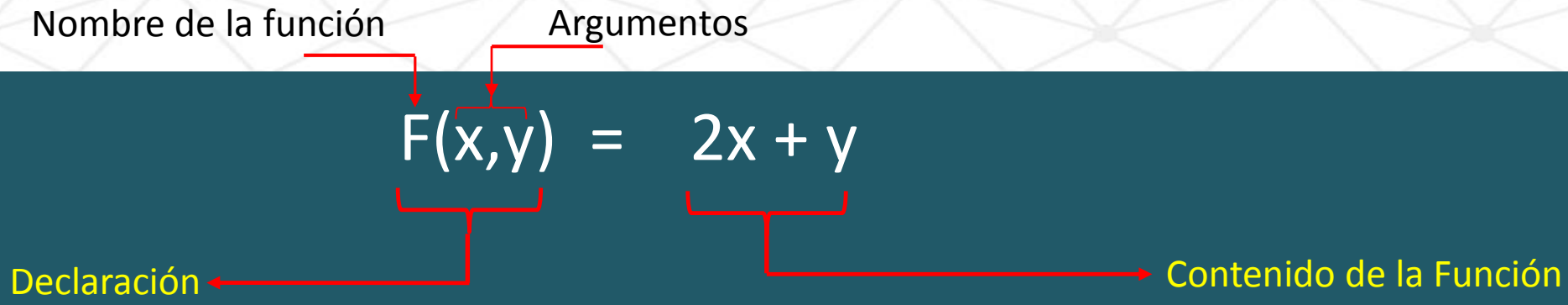
0.5 ← 4/F(3.5)

.

.

FUNCIONES - INTRODUCCIÓN

En matemática: **Ejemplo 2**



Aplicación de la función:

23 ← F(10,3)

9 ← 3 + F(1,7)

0.44 ← 4/F(3.5,2)

.

.

PROGRAMACIÓN - FUNCIONES CON RETORNO

Lenguaje C++:

Para el ejemplo 1:

```
#include <iostream>

using namespace std;

float F(float x)
{
   return 2*x+1;
}

int main()
{
      cout<<F(10)<<endl;
      cout<<3 + F(1)<<endl;
      cout<<4/F(3.5)<<endl;

      return 0;
}
```

Implementación de la función

Aplicación de la función:
La función es ***llamada*** o ***invocada*** en main()

Resultado de la ejecución

```
21
6
0.5

--------------------------------
Process exited after 0.2317 seconds with return value 0
Press any key to continue . . .
```

PROGRAMACIÓN - FUNCIONES CON RETORNO

Lenguaje C++:

```
#include <iostream>

using namespace std;

float F(float x, float y)
{
   return 2*x+y;
}

int main()
{
      cout<<F(10,3)<<endl;
      cout<<3 + F(1,7)<<endl;
      cout<<4/F(3.5,2)<<endl;

      return 0;
}
```

Implementación de la función

Para el ejemplo 2:

Aplicación de la función:
La función es ***llamada*** o ***invocada*** en main()

Resultado de la ejecución

```
23
12
0.444444

--------------------------------
Process exited after 0.1847 seconds with return value 0
Press any key to continue . . .
```

PROGRAMACIÓN - FUNCIONES CON RETORNO

Ejercicio:

Implementar una función que permita retornar un número aleatorio entero en el rango [a, b].

La función deberá ser llamada en main() para comprobar si retorna el valor esperado

¿Qué se espera de la función cuando sea invocada?

6 ← generaAleatorio(10, 20)

11 ← generaAleatorio(5, 12)

678 ← generaAleatorio(100, 999)

20 ← generaAleatorio(0, 30)

Solución:

```
int generaAleatorio(int a, int b)
{
    return rand()%(b-a+1)+a;
}
```

Implementación de la función

```
int main()
{
  srand(time(NULL));

  cout<<"Aleatorio en el rango [10,20]: "<<generaAleatorio(10,20);
  cout<<"\nAleatorio en el rango [5,12]: "<<generaAleatorio(5,12);
  cout<<"\nAleatorio en el rango [100,999]: "<<generaAleatorio(100,999);
  cout<<"\nAleatorio en el rango [0,30]: "<<generaAleatorio(0,30);

  return 0;
}
```

Invocación o llamada de la función
(para comprobar si retorna el resultado esperado)

Resultado de la ejecución

```
Aleatorio en el rango [10,20]: 15
Aleatorio en el rango [5,12]: 7
Aleatorio en el rango [100,999]: 650
Aleatorio en el rango [0,30]: 26
--------------------------------
Process exited after 0.1436 seconds with return value 0
Press any key to continue . . .
```

Ejercicio:

Implementar una función que permita retornar el factorial de un número.

La función deberá ser llamada en main() para comprobar si retorna el valor esperado

¿Qué se espera de la función cuando sea invocada?

1 ← factorial(1)

120 ← factorial(5)

5040 ← factorial(7)

3628800 ← factorial(10)

Solución:

```
double factorial(int n)
{
   double fact=1;

   for(int i=1;i<=n;i++)
   {
         fact = fact*i;
   }

   return fact;
}
```

Implementación de la función

```
int main()
{
      cout<<"Factorial de 1 es: "<<factorial(1)<<endl;
      cout<<"Factorial de 5 es: "<<factorial(5)<<endl;
      cout<<"Factorial de 7 es: "<<factorial(7)<<endl;
      cout<<"Factorial de 10 es: "<<factorial(10)<<endl;

      return 0;
}
```

Invocación o llamada de la función
(para comprobar si retorna el resultado esperado)

Resultado de la ejecución

```
Factorial de 1 es: 1
Factorial de 5 es: 120
Factorial de 7 es: 5040
Factorial de 10 es: 3.6288e+006

--------------------------------
Process exited after 0.1554 seconds with return value 0
Press any key to continue . . .
```

PROGRAMACIÓN - FUNCIONES SIN RETORNO

¡Una función NO siempre tiene que retornar un valor!

Ejemplo:

Implementar una función que imprima en pantalla un línea formada por N asteriscos

¿Qué se espera de la función cuando sea invocada?

imprimeLinea(10) → **********

imprimeLinea(5) → *****

imprimeLinea(50) → **********

*En este ejemplo **La función NO retorna** el valor de un dato, ya que sólo se encarga de la impresión en pantalla de un resultado*

Solución:

```
void imprimeLinea(int n)
{
   for(int i=1;i<=n;i++)
   {
        cout<<"*";
   }
}
```

Implementación de la función

```
int main()
{

    imprimeLinea(10);
    cout<<endl;

    imprimeLinea(5);
    cout<<endl;

    imprimeLinea(50);
    cout<<endl;

    return 0;
}
```

Invocación o llamada de la función
(para comprobar si retorna el resultado esperado)

Resultado de la ejecución

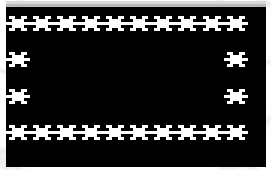
```
**********
*****
**************************************************

--------------------------------
Process exited after 0.1656 seconds with return value 0
Press any key to continue . . .
```

Problema

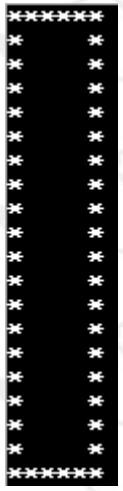
Implemente una función que permita dibujar en pantalla un rectangúlo de largo y altura enviados como argumentos. El rectángulo deberá dibujarse con caracteres *

¿Qué se espera de la función cuando sea invocada?

dibujaRectangulo(4,10)



dibujaRectangulo(6,20)



dibujaRectangulo(20,20)