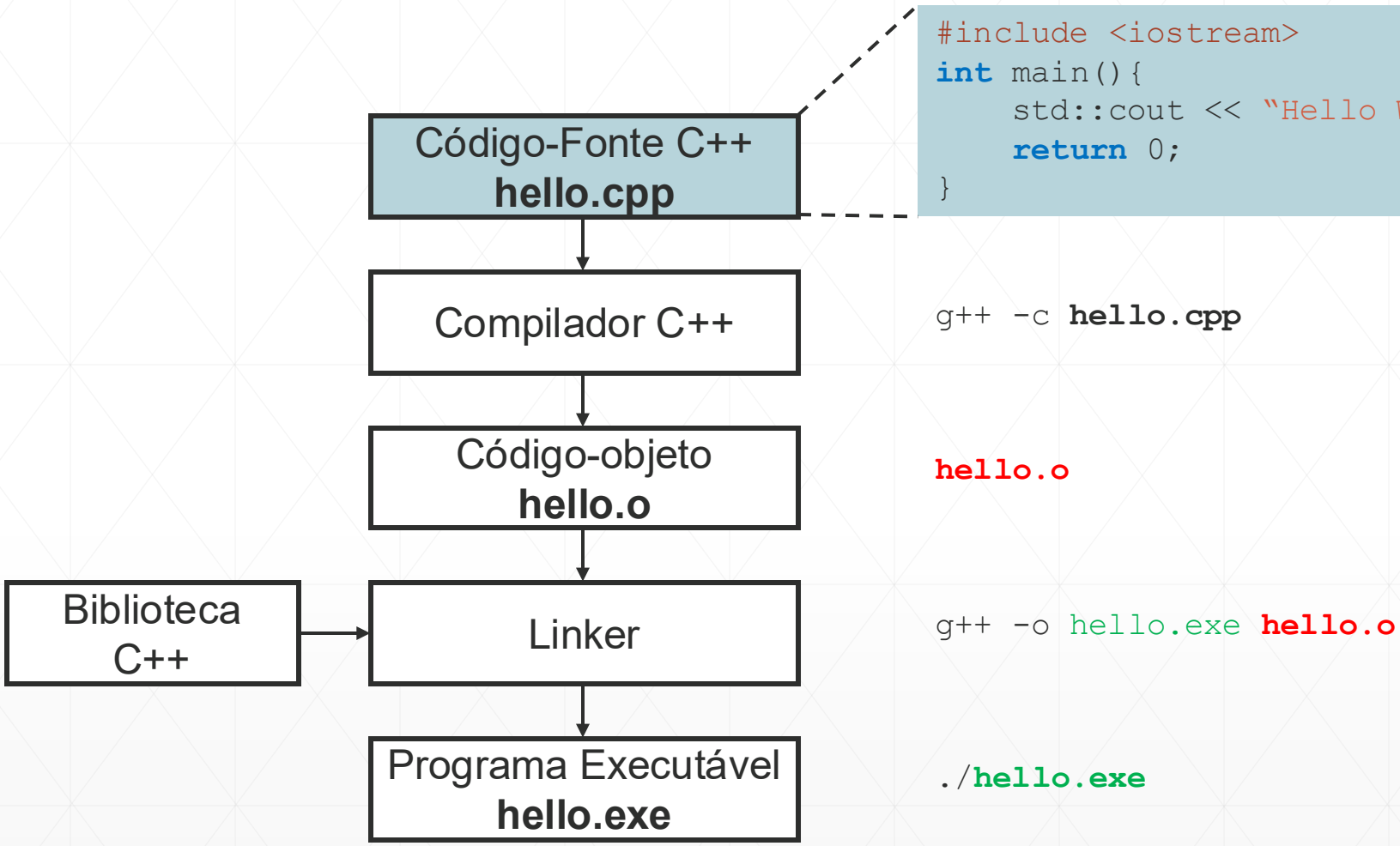
Técnicas de Programação

Prof. Protásio

Etapas da criação de um programa em C++




```
#include <iostream>
int main(){
    std::cout << "Hello World \n";
    return 0;
}
```

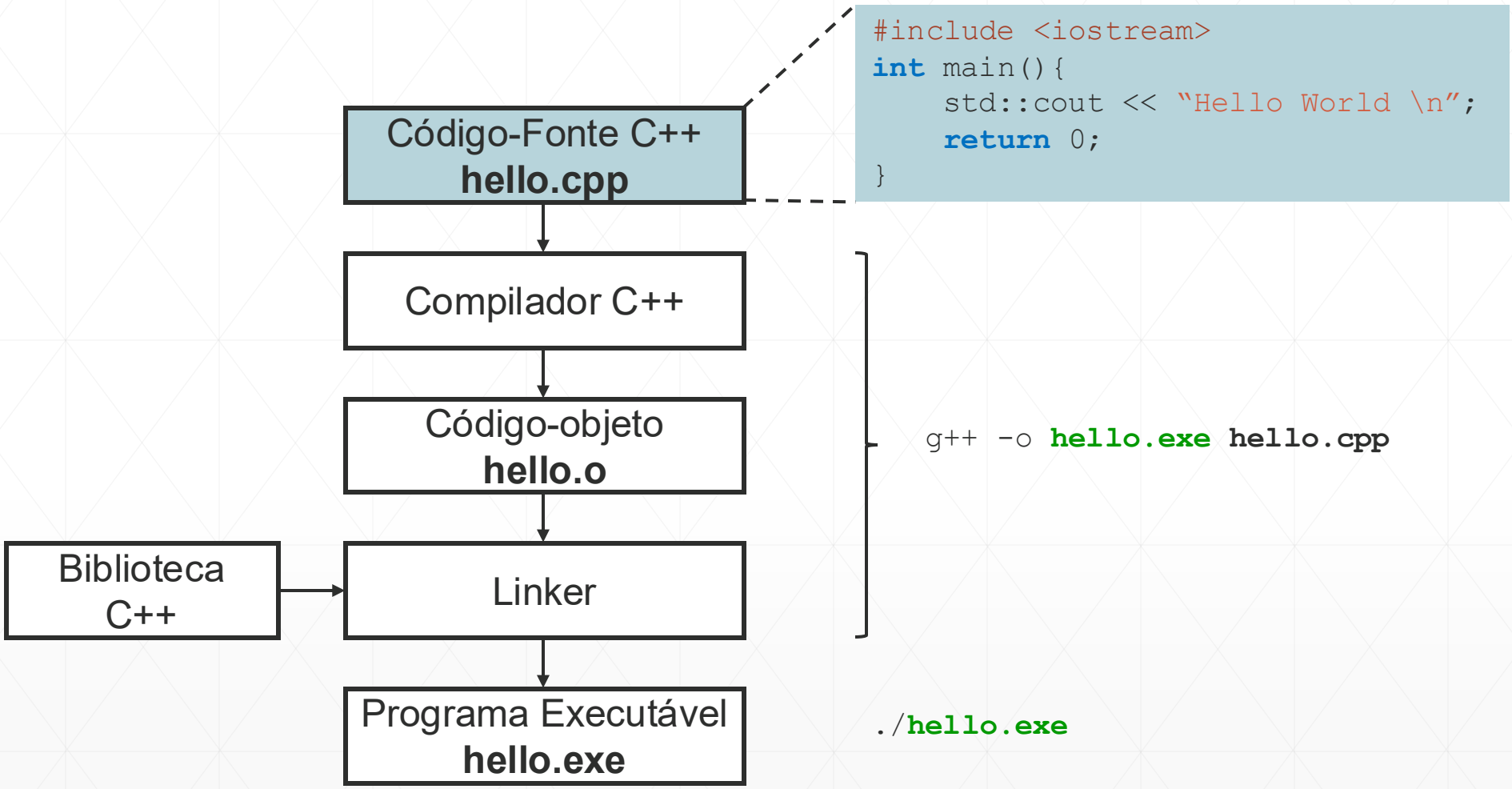
```
g++ -c hello.cpp
```

```
hello.o
```

```
g++ -o hello.exe hello.o
```

```
./hello.exe
```

Etapas da criação de um programa em C++



Estrutura típica de um programa C++

```
#include    Diretivas de pré-processamento
#define     Definição de macros
```

```
Declarações globais
    Variáveis Globais
    Protótipos de funções
```

```
Função principal main
    main()
    {
        Variáveis locais
        Instruções
                :
    }
```

```
Definição de outras funções
    func1(){
        ...
    }
    func2(){
        ...
    }
        :
```

```
#include <iostream>
#define  X  26

int Y = 4;

int media(int,int);


int main(){
    int Z = X + Y;
    Y = media(Z,2);
    std::cout << Y << "\n";
    return 0;
}


int media(int a, int b){
   std::cout << "media de";
   std::cout << a << "e" << b<< "é: \n";
   return (a + b)/2 ;
}
```

16

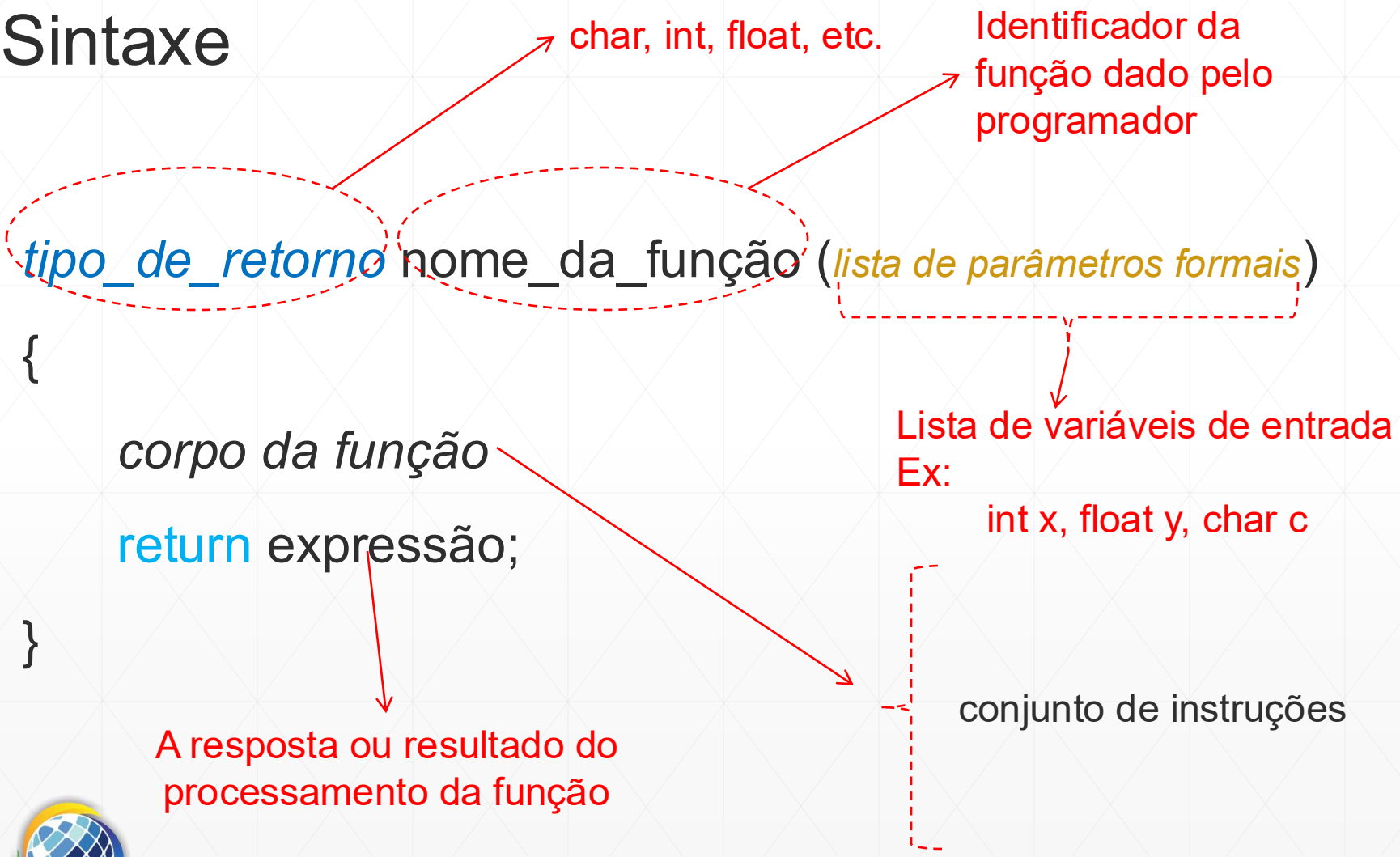
30

2

Funções

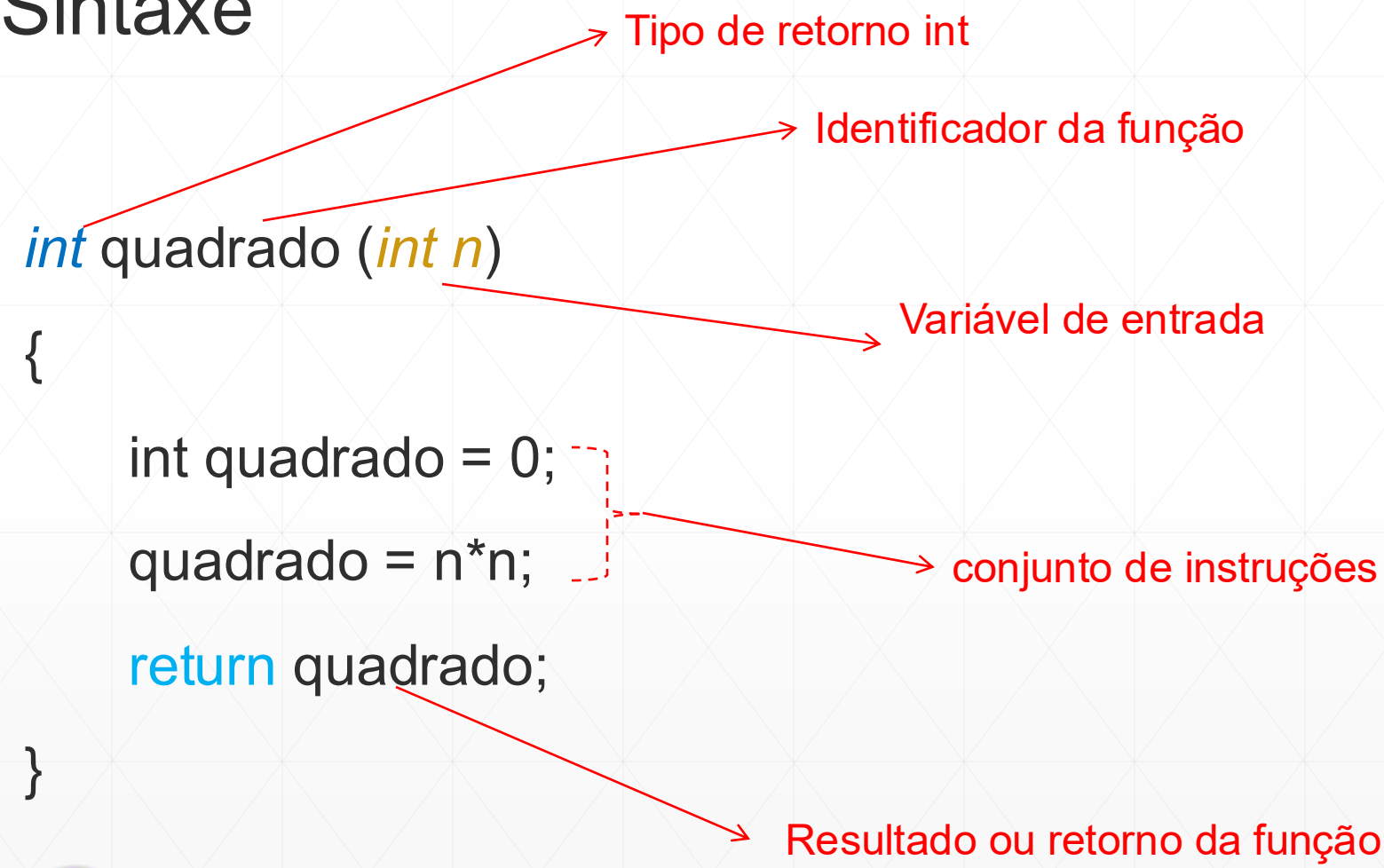
Sintaxe geral de Funções

- Sintaxe



Sintaxe geral de Funções

- Sintaxe



Tipos de funções

- **Funções definidas pelo usuário**
  - São as funções criadas pelo programador do código-fonte.
- **Funções de biblioteca**
  - São funções prontas para serem executadas e que vêm com a linguagem C++ (*C++ Standard Library*)

Tipos de funções

- **Funções definidas pelo usuário**
  - São as funções criadas pelo programador do código-fonte.
  - Exemplo:
    - Função `maior()` no código ao lado
      - Por que, na execução do programa, não é mostrado o resultado na tela?

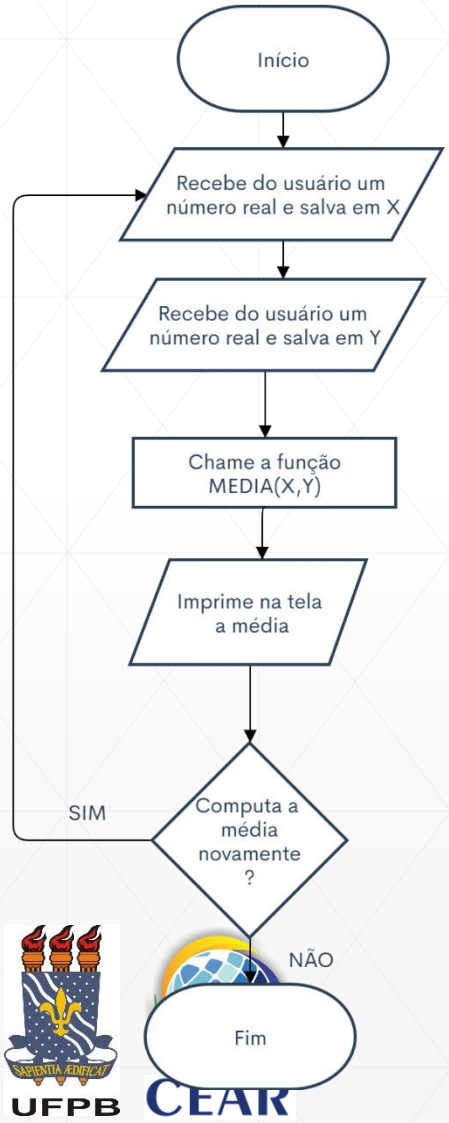
```
int maior(int, int); // Protótipo da função

int main(){
    int X = 1, Y = 3;
    int Z = maior(X, Y);
    return 0;
}

int maior(int a, int b){
    if (a > b)
        return a;
    else
         return b;
}
```

Situação-problema 1:

- Modifique o programa abaixo de forma a ter uma função que **retorne o menor valor** e mostrar na tela o resultado.

Funções definidas pelo usuário

- Cálculo de média aritmética




```cpp
#include <iostream>

float media(float, float); // Protótipo da função

int main(){

    float X, Y, Z;

    int SAIR = 0;  // SAIR = 0, continuar. SAIR = 1, sair do programa

    do{
        std::cout << " Digite X: "  << std::endl;
        std::cin >> X;

        std::cout << " Digite Y: "  << std::endl;
        std::cin >> Y;

        Z = media(X, Y);

        std::cout << " Média = " << Z << std::endl;

        std::cout << " SAIR = 0, continuar. SAIR = 1, sair do programa: "  << std::endl;
        std::cin >> SAIR;

    }while(SAIR == 0);

    return 0;
}

float media(float A, float B){// Corpo da função
    float M;
    M = A + B;
    M = M/2;
    return M;
}
```

Situação-problema:

- Modifique o programa de forma que a função media(,) tenha somente 1 linha de código.

Situação-problema:

- Modifique o programa de forma a receber 3 **três valores** e mostrar na tela a média desses valores.

Funções definidas pelo usuário

- Cálculo de média com valores dados pelo usuário

$$\text{arithmetic mean} = \frac{\sum_{n=1}^{k} x_n}{k}$$

```
#include <iostream>

void somatorio(float); // Protótipo da função

float  Soma_Atual = 0;
int    n = 0;

int main(){

    float x;

    do{
        std::cout << " Digite próximo VALOR: "  << std::endl;
        std::cin >> x;

         if (x == -999)
             continue;
         else
             somatorio(x);


        std::cout << " Digite -999 para mostrar média final: "  << std::endl;

    }while(x != -999); // Critério de Parada

    std::cout << " Média = " <<  Media_Atual / n << std::endl;

    return 0;
}

void  somatorio(float a){// Corpo da função
     Soma_Atual = Soma_Atual + a;        // Acumula
    n = n + 1;                           // Incrementa contador

}
```

Tipos de funções

- **Funções de biblioteca**
  - São **funções prontas e disponíveis** para serem utilizadas e que vêm com a linguagem C++
    - (*C++ Standard Library*)
    - A biblioteca padrão C++ prover 33 **cabeçalhos**:

<algorithm>	<iomanip>	<list>	<queue>	<streambuf>
<bitset>	<ios>	<locale>	<set>	<string>
<complex>	<iosfwd>	<map>	<sstream>	<typeinfo>
<deque>	<iostream>	<memory>	<stack>	<utility>
<exception>	<istream>	<new>	<stdexcept>	<valarray>
<fstream>	<iterator>	<numeric>	<strstream>	<vector>
<functional>	<limits>	<ostream>		

Tipos de funções

- Funções de biblioteca
  - Adicionalmente à biblioteca padrão C++, C++ também tem acesso à 18 cabeçalhos da biblioteca padrão C:

```
<cassert> <ciso646> <csetjmp> <cstdio>  <ctime>
<cctype>  <climits> <csignal> <cstdlib> <cwchar>
<cerrno>  <clocale> <cstdarg> <cstring> <cwctype>
<cfloat>  <cmath>   <cstddef>
```

Tipos de funções

- **Funções de biblioteca**
  - Biblioteca ***iostream***

Situação-problema 1:
- Modifique o programa para mostrar o quadrado de x (x^2), o cubo de x (x^3).

Lembre-se que em C, o include tem *.h
#include <stdio.h>

```
#include <iostream>

int main(){
    float x;
    float y;
    std::cin >> x >> y; // Recebe o valor de X e depois o de Y
    std::cout << "O valor de x eh: " << x << std::endl;
    std::cout << "O valor de y eh: " << y << std::endl;
    std::cout << "x + y = " << x + y << std::endl;
    std::cout << "x - y = " << x - y << std::endl;
    std::cout << "x * y = " << x * y << std::endl;
    std::cout << "x / y = " << x / y << std::endl;

    return 0;
}
```

>> OPERADOR DE EXTRAÇÃO
<< OPERADOR DE INSERÇÃO

Biblioteca <cmath>

float pow (float base, float exp);

$= base^{exp}$

- Função pow()

```
#include <iostream>
#include <cmath>

int main(){
    // typical usage
    std::cout << "pow(2, 10) = " << std::pow(2,10) << '\n'
              << "pow(2, 0.5) = " << std::pow(2,0.5) << '\n'
              << "pow(-2, -3) = " << std::pow(-2,-3) << '\n';
    // special values
    std::cout << "pow(-1, NAN) = " << std::pow(-1,NAN) << '\n'
              << "pow(+1, NAN) = " << std::pow(+1,NAN) << '\n'
              << "pow(INFINITY, 2) = " << std::pow(INFINITY,2) << '\n'
              << "pow(2,-INFINITY) = " << std::pow(2,-INFINITY) << '\n'
              << "pow(2,INFINITY) = " << std::pow(2,INFINITY) << '\n'
              << "pow(-INFINITY,1) = " << std::pow(-INFINITY,1) << '\n'
              << "pow(INFINITY, -1) = " << std::pow(INFINITY, -1) << '\n';
}
```

NAN = Not A Number
- 0/0 é NAN
- Log(- número) é NAN

NAN é uma forma da operação retorna erro

Biblioteca <cmath>

- Função sqrt()

```
float sqrt ( float arg );
```

$$= \sqrt{arg}$$

```
#include <iostream>
#include <cmath>

int main(){
    // normal use
    std::cout << "sqrt(100) = " << sqrt(100) << '\n'
              << "sqrt(2) = " << sqrt(2) << '\n'
              << "golden ratio = " << (1+sqrt(5))/2 << '\n';
    // special values
    std::cout << "sqrt(-0) = " << sqrt(-0.0) << '\n';
    return 0;
}
```

```
sqrt(100) = 10
sqrt(2) = 1.41421
golden ratio = 1.61803
sqrt(-0) = -0
```

Biblioteca <cmath>

float cbrt (float arg);

$= \sqrt[3]{arg}$

- Função cbrt()
  - Retorna a raiz cúbica

```
cbrt(729) = 9
cbrt(-0.125) = -0.5
cbrt(-0) = -0
cbrt(+inf) = inf
```

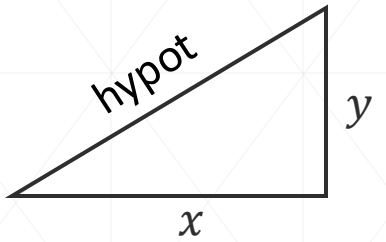
```
#include <iostream>
#include <cmath>

int main()
{
    // normal use
    std::cout << "cbrt(729) = " << cbrt(729) << '\n'
              << "cbrt(-0.125) = " << cbrt(-0.125) << '\n';
    // special values
    std::cout << "cbrt(-0) = " << cbrt(-0.0) << '\n'
              << "cbrt(+inf) = " << cbrt(INFINITY) << '\n';
    return 0;
}
```

Biblioteca <cmath>

float hypot (float x, float y);

$$\sqrt{x^2 + y^2}$$

- Função hypot()




```
#include <iostream>
#include <cmath>

int main()
{    float x,y,h;
     x = 3;
     y = 4;
     h = hypot(x,y);
     std::cout << "Hipotenusa igual a " << h << '\n';
     return 0;
}
```

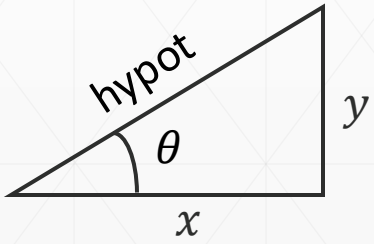
Situação-problema:

- Crie uma função definida pelo usuário que compute a hipotenusa como H=$\sqrt{X^2 + Y^2}$ e compare o resultado com o da função hypot.

Biblioteca <cmath>

- Algumas funções trigonométricas
  - Computa o seno, cosseno e tangente de theta, medido em **radiano**.

```
float sin ( float theta);
float cos ( float theta);
float tan ( float theta);
```




```
#include <iostream>
#include <cmath>

const double pi = acos(-1);

int main(){
    float x,y,h;
    x = 3;
    y = 4;
    h = hypot(x,y);
    std::cout << "Hipotenusa igual a " << h << '\n';
    std::cout << "Seno de Theta igual a " << y/h << '\n';
    std::cout << "Cosseno de Theta igual a " << x/h << '\n';
    std::cout << "Tangente de Theta igual a " << y/x << '\n';

    std::cout << "sin(pi/6) = " << sin(pi/6) << '\n'
              << "sin(pi/2) = " << sin(pi/2) << '\n'
              << "sin(-3*pi/4) = " << sin(-3*pi/4) << '\n';

    std::cout << "cos(pi/3) = " << cos(pi/3) << '\n'
              << "cos(pi/2) = " << cos(pi/2) << '\n'
              << "cos(-3*pi/4) = " << cos(-3*pi/4) << '\n';

    std::cout << "tan  (pi/4) = " << tan(  pi/4) << '\n' // 45 deg.
              << "tan(3*pi/4) = " << tan(3*pi/4) << '\n' // 135 deg
              << "tan(5*pi/4) = " << tan(5*pi/4) << '\n' // -135 deg
              << "tan(7*pi/4) = " << tan(7*pi/4) << '\n'; // -45 deg
    // special values
    std::cout << "sin(+0) = " << sin(0.0) << '\n'
              << "sin(-0) = " << sin(-0.0) << '\n';
    std::cout << "cos(+0) = " << cos(0.0) << '\n'
              << "cos(-0) = " << cos(-0.0) << '\n';
    std::cout << "tan(+0) = " << tan(0.0) << '\n'
              << "tan(-0) = " << tan(-0.0) << '\n';
    return 0;
}
```

Biblioteca <cmath>

- Arco seno
- Arco cosseno
- Arco tangente

```
float asin ( float arg);
float acos ( float arg);
float atan ( float arg);
```

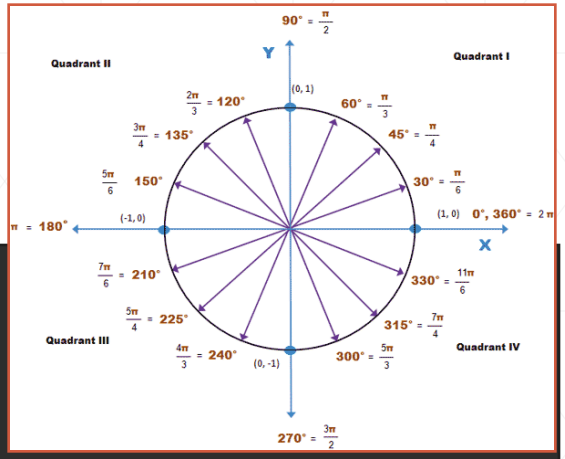
- Algumas funções trigonométricas

```
#include <iostream>
#include <cmath>

int main()
{   std::cout << "asin(1.0) = " << asin(1) << '\n'
              << "2*asin(1.0) = " << 2*asin(1) << '\n'
              << "asin(-0.5) = " << asin(-0.5) << '\n'
              << "6*asin(-0.5) =" << 6*asin(-0.5) << '\n';
     std::cout << "acos(-1) = " << acos(-1) << '\n'
              << "acos(0.0) = " << acos(0.0) << " 2*acos(0.0) = " << 2*acos(0) << '\n'
              << "acos(0.5) = " << acos(0.5) << " 3*acos(0.5) = " << 3*acos(0.5) << '\n'
              << "acos(1) = " << acos(1) << '\n';
    std::cout << "atan(1) = " << atan(1) << " 4*atan(1) = " << 4*atan(1) << '\n';

    // special values
    std::cout << "asin(0.0) = " << asin(0) << " asin(-0.0)=" << asin(-0.0) << '\n';
    std::cout << "acos(1.1) = " << acos(1.1) << '\n'; // outside the range [-1.0, 1.0]
    std::cout << "atan(Inf) = " << atan(INFINITY)
              << " 2*atan(Inf) = " << 2*atan(INFINITY) << '\n'
              << "atan(-0.0) = " << atan(-0.0) << '\n'
              << "atan(+0.0) = " << atan(0) << '\n';
    return 0;
}
```

Biblioteca <cmath>

```
float atan2 ( float y, float x );
```

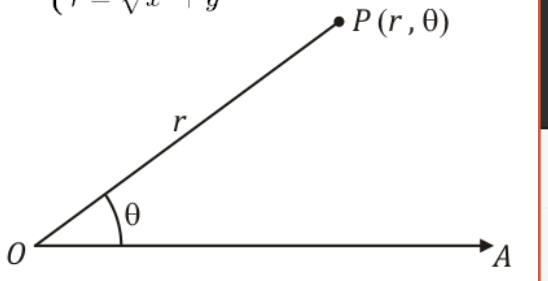
- Função arco tangente com 2 argumentos
  - Computa o arco tangente de $\frac{y}{x}$ usando os sinais desses argumentos para determinar o quadrante correto




```
#include <iostream>
#include <cmath>

int main(){
    // normal usage: the signs of the two arguments determine the quadrant
        // atan2(1,1) = +pi/4, Quad I
    std::cout << "(+1,+1) cartesian is (" << hypot(1,1)   << ',' << atan2(1,1)   << ") polar \n"
        // atan2(1, -1) = +3pi/4, Quad II
              << "(+1,-1) cartesian is (" << hypot(1,-1)  << ',' << atan2(1,-1)  << ") polar \n"
        // atan2(-1,-1) = -3pi/4, Quad III
        << "(-1,-1) cartesian is (" << hypot(-1,-1) << ',' << atan2(-1,-1) << ") polar \n"
        // atan2(-1,-1) = -pi/4, Quad IV
        << "(-1,+1) cartesian is (" << hypot(-1,1)  << ',' << atan2(-1,1)  << ") polar \n";
    // special values
    std::cout << "atan2(0, 0) = " << atan2(0,0)              // 0
              << " atan2(0,-0) = " << atan2(0,-0.0) << '\n'  // pi
              << "atan2(7, 0) = " << atan2(7,0)              // pi/2
              << " atan2(7,-0) = " << atan2(7,-0.0) << '\n'; // pi/2
}
```

(para qualquer valor, 7 foi um exemplo)

$$\begin{cases} \theta = \text{arctg}\left(\frac{y}{x}\right) \\ r = \sqrt{x^2 + y^2} \end{cases} \qquad \begin{cases} x = r\cos(\theta) \\ y = r\,\text{sen}(\theta) \end{cases}$$



Espaço de nomes

- É uma região declarativa que vincula um **identificador** a um grupo de variáveis, funções, subespaços, estruturas, classes, constantes etc.
- Exemplo:

```
namespace maria {
    int idade = 18;
    float cre = 8.3;
}
```

```
namespace jose {
  int idade = 21;
  float cre = 4.3;
}
```

  - No exemplo, `idade` e `cre` fazem parte do namespace `maria`
  - Espaços de nomes servem para evitar conflitos com nomes de variáveis iguais.

Espaço de nomes

- Qual a saída do programa abaixo:

```
#include <iostream>

namespace maria {
   int idade = 18;
   float cre = 8.3;
}

namespace jose {
   int idade = 21;
   float cre = 4.3;
}

int main(){
     int idade = 12;
     std::cout << idade << " " << maria::idade << std::endl;
     std::cout << jose::idade << std::endl;
     return 0;
}
```

:: operador de resolução de escopo

Espaço de nomes

- Qual a saída do programa abaixo:

```
#include <iostream>

using namespace std;

namespace maria {
   int idade = 18;
   float cre = 8.3;
}

namespace jose {
   int idade = 21;
   float cre = 4.3;
}

int main(){
    int idade = 12;
    cout << idade << " " << maria::idade << endl;
    cout << jose::idade << endl;
    return 0;
}
```

- Um espaço de nome bem útil e já definido na biblioteca padrão C++ é:
  std definido na biblioteca iostream
- Para usa este espaço de nome:
  `using namespace std;`
- `cout e endl estão em std.`

Espaço de nomes

- Definindo uma função em um espaço de nome

```
#include <iostream>

using namespace std;

namespace maria {
   int idade = 18;
   float cre = 8.3;
   void incrementar_idade() { idade++; }
}

namespace jose {
   int idade = 21;
   float cre = 4.3;
}

int main(){
    cout << maria::idade << endl;
    maria:: incrementar_idade();
    cout << maria::idade << endl;
    return 0;
}
```

Situação-problema:

- Modifique o programa a fim de ser possível incrementar a idade de jose e mostrar na tela.

Lista de Exercícios

- Faça um programa em C++ que realize em sequência as seguintes ações sempre após a digitação de qualquer tecla (se o usuário digitar S, o programa deve ser encerrado):

1. Imprima na tela seu nome completo e, em seguida, todos os valores pares entre 0 e sua idade.
2. Usando funções da biblioteca padrão c++:
   A. Imprima a raiz quadrada de todos os **números pares** de 1 até 100.
   B. Imprima a hipotenusa, o perímetro, a área, o seno, cosseno e tangente dos ângulos de um triângulo retângulo cujos catetos sejam dados pelo usuário.
3. Imprima o **diâmetro**, o **perímetro** e a **área** de um círculo cujo raio seja dado pelo usuário. Utilize o operador `#define` para definir o valor de pi.
4. Mostre na tela os valores dos senos e cossenos dos ângulos 0º até 90º com passo de 5º.

Literatura pesquisa para esta aula

- Livro-texto
- https://en.cppreference.com/
- http://www.cplusplus.com