

# Linguagem de Programação

Funções

**ECT2303**

helton.maia@ect.ufrn.br

# Funções

- **Grupo de instruções** que de forma conjunta executam uma **tarefa particular**;
- Considere uma função para **somar dois números**:
  - A função recebe um **nome** e através deste, pode ser ativada.
  - A função pode **retornar** um valor (ex: A soma de 3 e 2 retorna 5).
  - A função pode receber **parâmetros de entrada**.

# Funções

- Todo programa C ++ tem pelo menos uma função, que é a `main()`, e pode também definir funções adicionais;
- Você pode dividir seu código em funções separadas, geralmente cada função executa uma tarefa específica.

# Funções

- Uma declaração de função informa ao compilador, o nome da função, seu tipo de retorno e parâmetros. Uma definição de função fornece o corpo real da função;
- A biblioteca padrão C++ fornece inúmeras funções internas que seu programa pode chamar. Por exemplo, tem-se o cabeçalho `<iostream>` como responsável pela manipulação do fluxo de dados padrão do sistema (entrada/saída padrão e saída de erros).

# Motivação para utilização de funções

- **Organização:** Permite reduzir um programa complicado em partes menores e mais gerenciáveis, reduzindo sua complexidade;
- **Reutilização de *software*:** Uma vez que uma função é escrita, ela pode ser chamada várias vezes dentro do programa, e ainda, ser compartilhada com outros.
- **Testes:** Reduzem a redundância, gerando menos código para testar. Se não forem modificadas, não é necessário a repetição de testes do seu funcionamento
- **Abstração:** Para utilizar uma função, você precisa saber o seu nome, entradas e saídas, não precisando saber detalhes de seu funcionamento.

# Exemplos de funções conhecidas

## <cmath>

### Trigonometric functions

<b>cos</b>	Compute cosine (function )
<b>sin</b>	Compute sine (function )
<b>tan</b>	Compute tangent (function )
<b>acos</b>	Compute arc cosine (function )
<b>asin</b>	Compute arc sine (function )
<b>atan</b>	Compute arc tangent (function )
<b>atan2</b>	Compute arc tangent with two parameters (function )

Fonte: <http://www.cplusplus.com/reference/cmath/>

# Funções (exemplo)

Como chamar  
uma função no  
seu programa?

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

using namespace std;

int main(){

    double x = 3.0;
    double y = pow(x, 2.0);

    cout << "x=" << x << " y=" << y << endl;
    return 0;

}
```

# Elementos de uma Função

## **Variáveis locais:**

- Variáveis declaradas nas definições de uma função;
- São conhecidas apenas na função em que são definidas;
- Declaradas no bloco de código pertencente à função;
- Criadas na memória no momento em que a função é chamada, sendo eliminadas ao final da execução desta função;

## **Parâmetros (ou lista de parâmetros):**

- Meio de transmissão de informação entre as funções;
- Parâmetros são tratados como variáveis locais de uma função;



# Definição de uma função

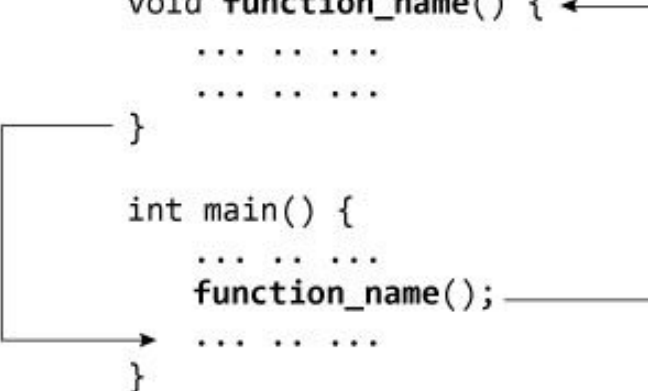
## Protótipo/Declaração/Assinatura

<tipo\_retorno> <nome>(lista de parâmetros);

## Implementação

```
<tipo_retorno> <nome>(lista de parâmetros){  
    declaração de variáveis  
    instruções  
}
```

```
#include <iostream>  
  
void function_name() {  
    ... ..  
    ... ..  
}  
  
int main() {  
    ... ..  
    function_name();  
    ... ..  
}
```



# Exemplo de função

Função para calcular uma **soma entre dois números** e retornar seu resultado.

```
// definicao da Funcao
int soma(int a, int b){
    int sum;
    sum = a + b;

    return sum;
}
```

# O comando **return**

- **Possui as seguintes propriedades:**

- Converte de forma automática o resultado da expressão para o tipo definido na função;
- Retorna um resultado;
- Termina a execução da função e retorna à sequência de instruções do código de chamada;

```
int prod(int a, int b){  
    return a*b;  
}
```

- Funções do tipo **void** não precisam utilizar o comando **return**.

```
void imprimir( int x ) {  
    cout << x ;  
}
```

# Observações

- De forma geral, o nome de uma **função pode aparecer** nos seguintes lugares de seu programa:
  - **Declaração** (protótipo/assinatura)
  - **Definição**
  - **Chamada**

**Obs:** Uma função não pode ser chamada sem antes ter sido declarada.  
Não deixe de escrever seu protótipo!

## Exemplo de função

Escreva um programa no qual exista uma função para calcular o cubo de um inteiro dado como argumento. Para testes, calcule o cubo de todos os números entre 1 e 10, inclusive, e imprima na tela os resultados.

# Exemplos de funções (solução)

```
#include <iostream>
using namespace std;

int cubo(int);

int main(){
    for(int i=1; i<=10; i++)
        cout << "Cubo de " << i << " = " << cubo(i) << endl;
    return 0;
}

int cubo(int x){
    int c = x*x*x;
    return c;
}
```