

Linguagem de Programação

Funções

ECT2303

helton.maia@ufrn.br

Funções em C++

- Uma função é um bloco de código (conjunto de instruções) que executa uma tarefa específica e pode ser chamado de diferentes partes do programa.
- As funções ajudam a evitar a duplicação de código e tornam o código mais organizado e fácil de entender e manter.

Funções em C++: Observações

- Todo programa C ++ tem pelo menos uma função, que é a `main()`. Além disso, pode-se definir funções adicionais;
- É recomendado que você divida o código em funções distintas, onde cada função vai executar uma tarefa específica;
- Uma declaração de função informa ao compilador, o nome da função, seu tipo de retorno e alguns parâmetros.
- A definição de função fornece o corpo real da função;

Funções em C++: Motivação para Uso

- **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 sistemas.
- **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.

Definição de uma função

Protótipo/Declaração/Assinatura

<tipo_retorno> <nome>(lista de parâmetros);

Implementação

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

Elementos de uma Função

Parâmetros (ou lista de parâmetros):

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

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;

Exemplo de função

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

```
// definição da Função
int soma(int a, int b){
    int sum;
    sum = a + b;

    return sum;
}
```


Funções em C++

A seguinte função em C++ recebe dois números inteiros como entrada e retorna o seu produto:

```
int multiplicacao(int a, int b) {  
    int produto = a * b;  
    return produto;  
}
```

Funções em C++

Neste exemplo, a seguinte função em C++ recebe dois números inteiros como entrada e retorna o seu produto:

```
int multiplicacao(int a, int b) {  
    int produto = a * b;  
    return produto;  
}
```

Para chamar a função `multiplicacao` em outro lugar do programa, basta fornecer os argumentos apropriados:

```
int resultado = multiplicacao(3, 4);
```

Funções em C++

Ao chamar uma função em C++, é possível passar valores como argumentos para a função. Existem dois métodos principais de passagem de argumentos: **por valor e por referência**.

```
void adicionar_um(int num) {  
    num = num + 1;  
}
```

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;  
}
```

Tipo sem retorno

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

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

Funções em C++

A **C++ Standard Library** (biblioteca padrão), fornece uma rica coleção de funções para realizar os cálculos matemáticos mais comuns, manipulação de string/caracter, entrada/saída, verificação de erros e outros.

C++ Standard Library: arquivos de cabeçalho

<code><iostream></code>	Contém protótipos para as funções de entrada/saída de fluxo
<code><iomanip></code>	Contém protótipos de funções para manipuladores de fluxo
<code><cmath></code>	Contém protótipos para as funções da biblioteca de matemática

Funções da biblioteca de matemática <cmath>

Trigonometric functions

cos

Compute cosine (function)

sin

Compute sine (function)

tan

Compute tangent (function)

acos

Compute arc cosine (function)

asin

Compute arc sine (function)

atan

Compute arc tangent (function)

atan2

Compute arc tangent with two parameters (function)

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;
    return 0;
}
```

Exercício

Escreva um programa que possua uma função capaz de calcular o cubo de um número inteiro (dado como argumento de entrada). Para os testes, calcule o cubo de todos os números entre 1 e 10, inclusive, e imprima na tela os resultados.

Exercício: Solução

```
// Função para calcular o cubo de um número
int calcularCubo(int num) {
    return num * num * num;
}

int main() {
    //Loop para calcular e imprimir o cubo dos números de 1 a 10
    for (int i = 1; i <= 10; i++) {
        int cubo = calcularCubo(i); // Chama a função cubo
        cout << "O cubo de " << i << " é: " << cubo << endl;
    }
    return 0;
}
```

?