



Linguagem C++

Funções – Aula I

Definição e exemplos

Prof. Bruno E. G. Gomes
IFRN





- Funções são blocos de código em um programa que executam uma tarefa bem definida
 - Ex.: somar, executar o programa (função *main*), buscar um valor em uma lista, etc.

- Facilitam a tarefa de programação
 - Cada parte do programa (função) pode ser desenvolvida e testada individualmente



- Uma função recebe dados (entrada) e devolve zero ou um dado de retorno para quem a chamou.
- Exemplos:
 - $1 + 2 = 3$
 - O operador soma recebe dois *operandos* e retorna um resultado
 - **int** res = 1 + 2;
 - o resultado é armazenado em res
 - **int** res = soma (1, 2);
 - uso de uma função soma definida pelo usuário

- Utilizamos diversas funções em C++
 - strlen, strcmp
 - funções de entrada/saída
 - *main*
- Um programa C++ estruturado consiste em:
 - utilizar as funções existentes na biblioteca C++
 - desenvolver e utilizar novas funções quando necessário



```
<tipo_retorno> <nome> (<p1, p2, ..., pn>) {  
    <instruções>  
  
    return <expressão>  
}
```

<tipo_retorno> - tipo de retorno da função (quando houver);

<nome> - nome da função. Utilizado para chamá-la, juntamente com as entradas passadas à função;

<p1, p2, ..., pn> - valores recebidos pela função (0 ou vários), ou parâmetros

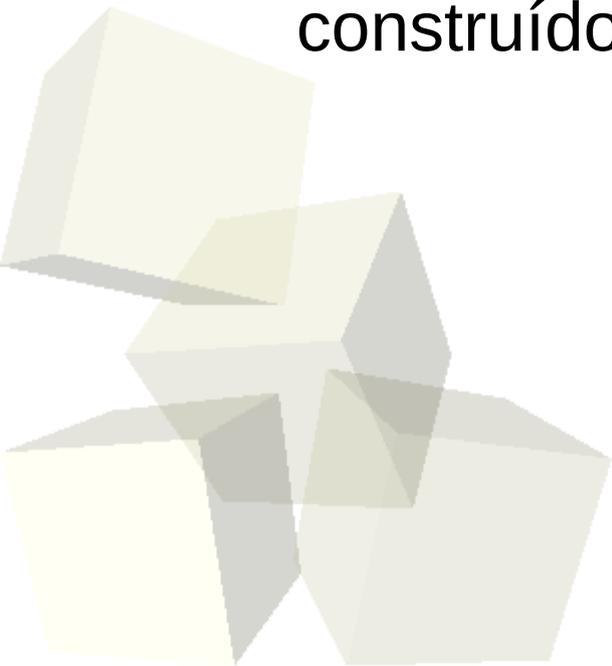
<instruções> - instruções que fazem parte da implementação da função;

return <expressão> - retorno da função, quando houver. Deve ser uma expressão do mesmo tipo que **<tipo_retorno>**.





- Número de valores recebidos (parâmetros da função):
 - 0 (zero) a vários
- Número de valores retornados:
 - 0 ou 1
- Tipo dos valores recebidos/retornados:
 - Qualquer tipo válido C++, seja ele primitivo ou construído pelo usuário





Exemplo: função quadrado

*Recebe um número inteiro no parâmetro x e retorna o quadrado deste número $x * x$*

tipo de retorno

nome

Tipo e nome do parâmetro de entrada

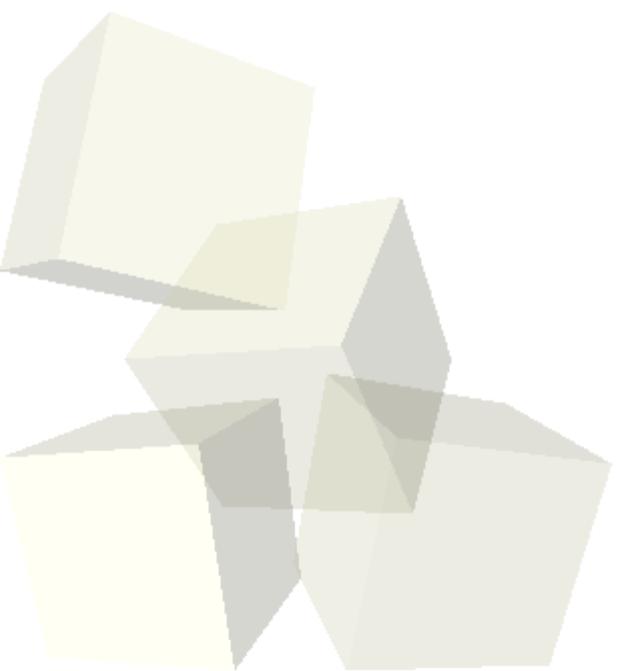
```
int quadrado (int x) {  
    return x * x;  
}
```

valor retornado pela função





- Crie o código para a função **impar**, que recebe um número inteiro e retorna se ele é ou não ímpar





- Após ser declarada, uma função pode ser utilizada por qualquer outra função no programa
- Para usar uma função:
 - Coloque o nome da função seguido da lista de valores recebidos
 - Os valores recebidos devem ser do mesmo tipo declarado na função
- Após o retorno, a execução da função é finalizada
 - ◆ A função irá retornar um valor (***return*** <expressão>) para o mesmo lugar onde ela foi chamada



- Chamando a função **quadrado** em *main*

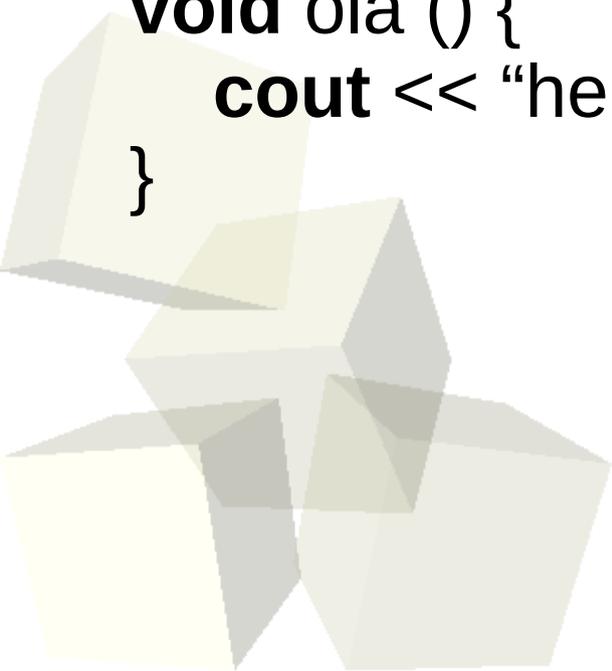
```
int main() {  
    int resultado;  
  
    /*2 é o argumento para a função. Pode ser qualquer  
    expressão inteira*/  
    resultado = quadrado(2);  
    cout << "O quadrado de" << 2 << "é" << quad;  
  
    return 0;  
}
```





```
//Uma função com mais de um parâmetro  
int produto (int x, int y)  
{  
    return x * y;  
}
```

```
//Uma função sem retorno e sem parâmetros(void)  
void ola () {  
    cout << "hello world!";  
}
```



- 1) Crie o código para a chamada à função **produto** na função *main*.
- 2) Declare a função **ehVogal** que recebe um caractere e diz se o valor recebido é ou não uma vogal.
- 3) Crie uma função **primo**, que recebe um inteiro e diz se ele é ou não primo.
- 4) Faça uma função **media_final** que recebe as notas dos 4 bimestres de um aluno do integrado do IFRN e calcula a sua média final.



- O código no corpo de uma função é privativo a ela
 - não é conhecido e não pode ser acessado por nenhuma outra função
- São variáveis locais à função
 - Os parâmetros da função (variáveis declaradas na lista de parâmetros)
 - Todas as variáveis declaradas em seu corpo (entre { e } da função)
- Deve-se evitar o uso de variáveis globais em uma função