

# Programação de computadores II - Revisão

## C++ Funções, Arrays, Structs

Elton M. Cardoso<sup>1</sup>

<sup>1</sup>Departamento de Ciências da Computação

22 de março de 2011

# Função

## Função

Função é uma sequência de instruções, usualmente corriqueiras, que é abstraída por um nome e um conjunto de parâmetros e pode ou não retornar algum resultado.

# sintaxe

```
tipo_de_retorno ident(t1 arg1 ,... ,tn arn){  
    corpo da funcao  
    return ..;  
}
```

Exemplos:

```
#include <iostream>
```

```
using namespace std;
```

```
int fahrToCelcius(int fahr){  
    int c;  
    c = (fahr - 32) * 5/9  
    return c;  
}
```

```
int main{  
    ....  
}
```

No momento da chamada o argumento passado para a função é copiado para a variável declarada como parâmetro. Os parâmetros são variáveis locais na função e apenas existem enquanto a função existe.

Quando o comando `return` é chamado, o sistema devolve o controle a quem chamou a função.

Funções podem não receber ou retornar parâmetros Que  
utilidade tem uma função destas ?

passagem de parâmetros por referência.

- Ao contrário da passagem por valor, a passagem por referência recebe um endereço de memória do parâmetro
- Permitem *side effects*
- Para denotar que um dado parâmetro é passado por referência usa-se o operador & (e-comercial) em logo após o tipo. Ex.: `int&`

Exemplo no quadro



# Arrays

Arrays (ou vetores) são um conjunto de dados, de uma mesmo tipo, organizados em uma região contígua de memória.

Aplicações: Considere o seguinte problema  
Calcular a média de uma turma de  $n$  alunos e imprimir as  
notas acima da média.

## Resolução no quadro

O tamanho de um array deve ser conhecido previamente a sua instanciação. Mas e se iss não for possíve ?

Duas abordagens principais:

- Usar um vetor de tamanho arbitrariamente grande.
- Alocar dinamicamente o vetor.

# Structs

Tipos de dados heterogênicos.

# Sintaxe

```
struct Nome{  
    tipo identificador;  
    tipo identificador;  
    .....  
};
```