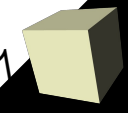




Fundamentos de Programação

Linguagem C++ Registros

Prof.: Bruno E. G. Gomes

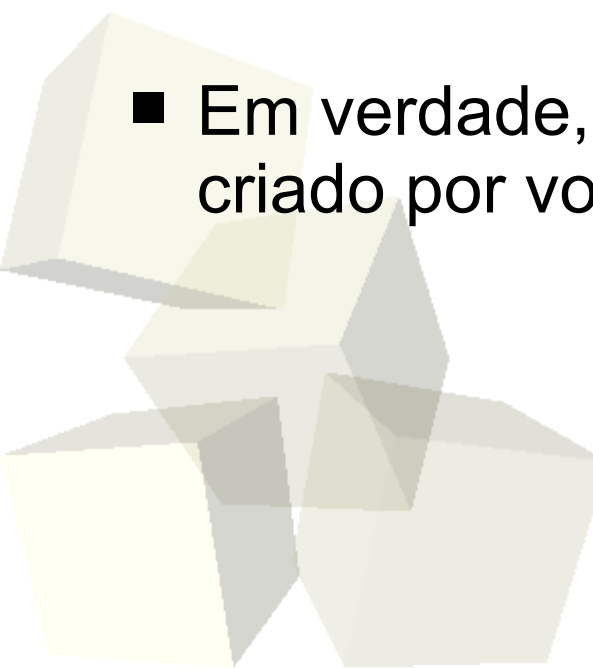




- Até o momento trabalhamos com variáveis de um mesmo tipo (Int, bool, float, etc.)
 - Um valor apenas (variável simples)
 - Vários valores relacionados a um mesmo nome de variável (vetores e matrizes)
- Por vezes é necessário trabalhar com tipos de dados compostos, por exemplo:
 - Um formulário de cadastro que contém informações de vários tipos diferentes
 - Dados de um funcionário, uma pessoa, etc.
- *Como representaríamos esse tipo de dados utilizando apenas variáveis de tipos básicos?*



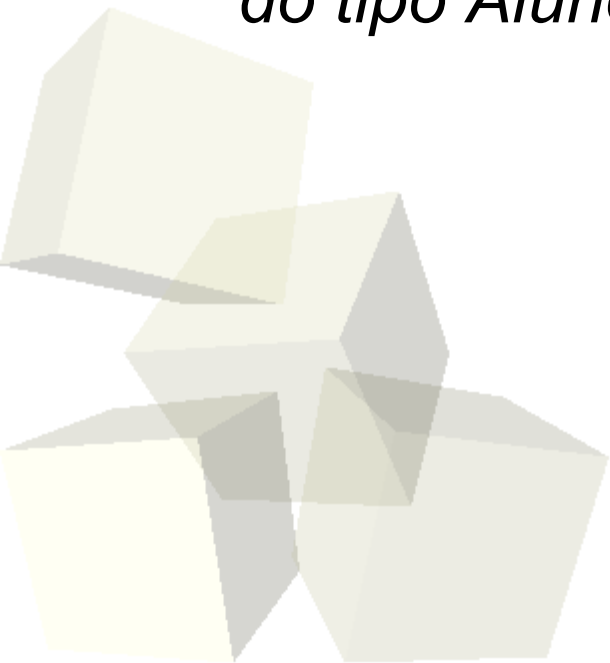
- Um meio de criar variáveis compostas é através de registros
- Um registro relaciona uma ou mais variáveis (campos do registro)
 - Variáveis podem ser de tipos distintos: básicos, vetores, outros registros, etc.
- Em verdade, um registro define um novo *tipo de dados* criado por você





Exemplo: tipo Aluno

- Um aluno possui diversas informações, tais como: matrícula, nome, curso, idade, cpf e telefone
- Usando registros é possível criar esse o tipo de dados *Aluno*
 - *Esse novo tipo pode ser então utilizado para criar variáveis do tipo Aluno na sua aplicação*





Criação do tipo Aluno

```
struct Aluno {  
    int matricula;  
    string nome;  
    string curso;  
    int cpf;  
    int telefone;  
};
```

- A estrutura Aluno define o novo tipo, composto pelas variáveis:
 - *matricula, nome, curso, cpf e telefone*





Uso do tipo Aluno

- É necessário apenas declarar variáveis do tipo Aluno
- Exemplo:
 - Aluno al1;
 - Aluno al3, aluno, al;
 - Aluno alunos[30];
- Para acessar o valor de cada campo de uma variável Aluno:

nome_da_variável.nome_campo

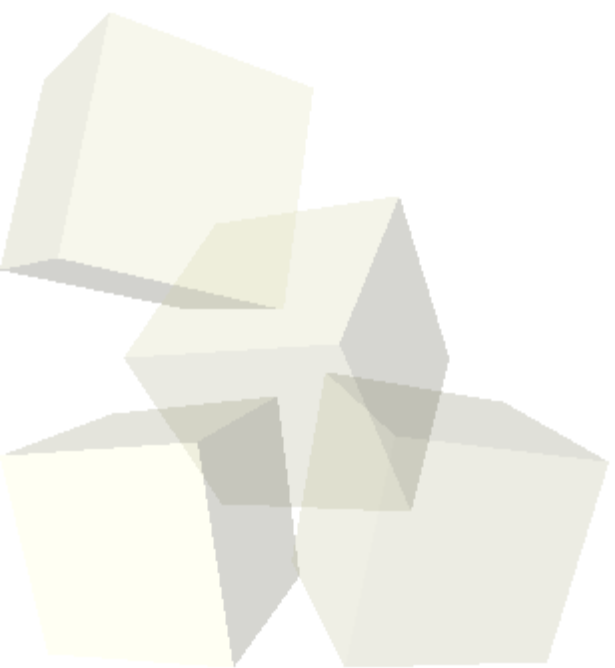
Exemplo: al1.nome = “José Francisco”;
cout << al1.nome;





Declaração de Struct – forma geral

```
struct <Nome> {  
    <tipo> <nome_var_1>;  
    <tipo> <nome_var_2>;  
    <tipo> <nome_var_3>;  
    <tipo> <nome_var_n>;  
};
```





Exemplo Aluno

```
#include <iostream>
#include <string>
```

```
using namespace std;
```

```
struct Aluno {
    int matricula;
    string nome;
    string curso;
    int cpf;
    int telefone;
};
```

```
int main () {
    Aluno al;
    al.matricula = 1234;
    al.nome = "Chico";
    al.curso = "Informática";
    al.cpf = 445454343;
    al.telefone = 32323232;

    cout << al.matricula;
    cout << al.nome;

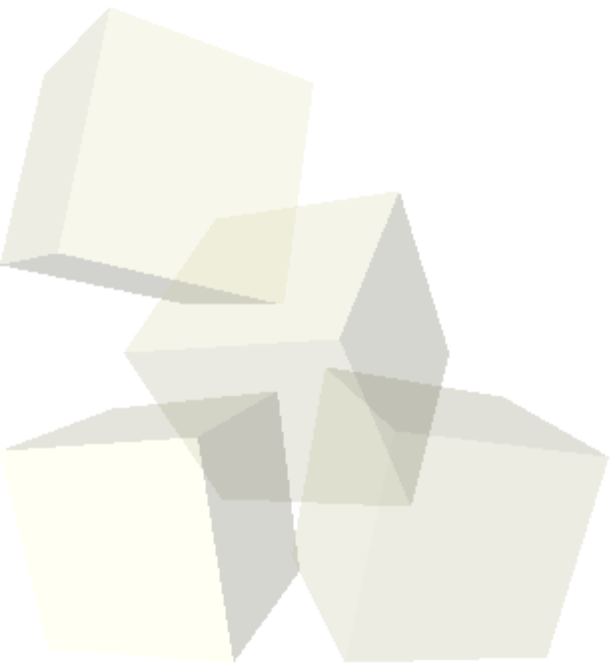
    return 0;
}
```





Estruturas e funções

- Na declaração da função:
 - Crie uma variável do tipo da estrutura
 - OBS.: A estrutura deve ser definida antes da assinatura da função.
- No uso:
 - Forneça o nome da variável estrutura





- Função para imprimir as informações de um aluno

```
void imprimirAluno (Aluno al) {  
    cout << "Nome: "      << al.nome      << endl;  
    cout << "CPF: "      << al.cpf       << endl;  
    cout << "Matrícula: " << al.matricula << endl;  
    cout << "Curso: "    << al.curso     << endl;  
    cout << "Telefone: " << al.telefone << endl;  
}
```

Código do exemplo completo [aqui](#)

