



Linguagem C++

Estruturas de controle – Parte I – decisão

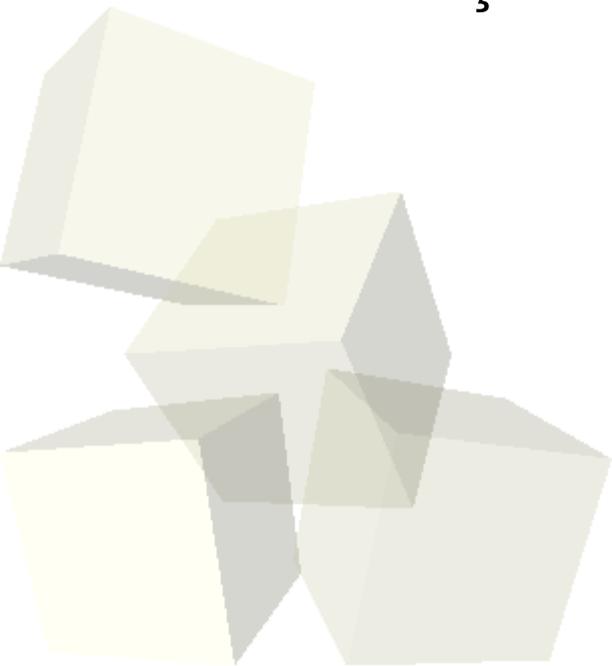
Prof. Bruno E. G. Gomes





Estruturas de Controle

- Permitem o controle da sequência de execução de um programa
- Em C++ temos:
 - ◆ Estruturas de decisão (ou seleção) – aula de hoje
 - ◆ Estruturas de repetição
 - ◆ Instruções para interrupção e desvio





Estrutura de Seleção IF (SE)

- Permite a escolha de qual bloco de comando será executado a partir da avaliação de uma condição
- Sintaxe:

```
if (<condição>) {  
    <comandos>  
}
```

Onde:

<condição> é qualquer expressão lógica ou aritmética e <comandos> é um bloco de uma ou mais instruções

Observação: Em C/C++, uma condição verdadeira pode ser qualquer valor diferente de zero. E uma condição falsa é representada por zero.





Estrutura de Seleção IF com ELSE (senão)

Sintaxe:

```
if (<condição>) {  
    <comandos_V>  
} else {  
    <comandos_F>  
}
```

Semântica (significado): Se a condição for verdadeira, então o bloco <comandos_V> é executado. Caso contrário (senão), o bloco <comandos_F> é executado.

Exemplo: [par1.cpp](#)





IFs aninhados (if-else-if)

USO: Quando várias testes tem que ser feitos, mas apenas 1 deles deve ser executado.

```
if (<expressão>) {  
    <comandos>  
} else if (<expressão>) {  
    <comandos>  
} else if (<expressão>) {  
    <comandos> ...  
} else {  
    <comandos>  
}
```

- As expressões são avaliadas a partir do primeiro **if**.
- Quando uma delas for verdadeira, o bloco do **if** correspondente é executado.
- Caso todas sejam falsas, o bloco do **else** é executado.
- Exemplo: [operacoes-if.cpp](#)





O operador ?

- Pode ser usado como um atalho para um IF do tipo:

```
if (<expressão>) {  
    <comandos_V>  
} else {  
    <comandos_F>  
}
```

Sintaxe:

<expressão> ? *<comandos_V>* : *<comandos_F>*

Exemplos: resultado = a > 0 ? c++ : c--
par2.cpp





Estrutura SWITCH (ESCOLHA)

- Testa o valor de uma expressão com relação a outros valores previamente conhecidos
 - ♦ Dentre todos os testes (comandos **case**) apenas 1 deve ser executado (caso algum seja verdadeiro)
 - ♦ Se nenhum teste for verdadeiro, os comandos em default são executados

- **Sintaxe:** *switch*(<expressão>) {

- case* <constante_1>:

- <comandos>

- break*;

- case* <constante_n>:

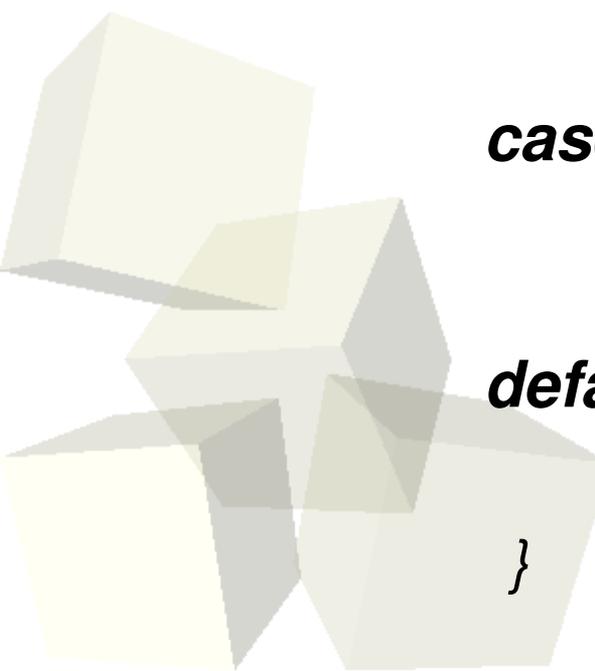
- <comandos>

- break*;

- default*:

- <comandos>

- }





Observações:

- O comando **switch** só pode testar expressões quanto à igualdade (==)
- Duas constantes **case** no mesmo **switch** não podem ter valores idênticos
- Normalmente, a **expressão** de teste no **switch** é uma variável ou constante inteira ou caractere
- comandos **break** são usados para sair da estrutura quando um caso foi executado:
 - ◆ Evita-se que se testem (e se executem) outros casos



- Exemplo: `operacoes.cpp`

- Exemplo de caso sem comandos:

```
switch(i) {  
    case 1:                                case 5:  
    case 2:                                num = 3;  
    case 3:                                }  
        num = 0;  
        break;  
    case 4:  
        num = 1;  
        break;
```