



Estruturas Condicionais II

Op. ternário



INF110 – Programação I

Prof. Alcione/André Gustavo
DPI/UFV – 2020/1

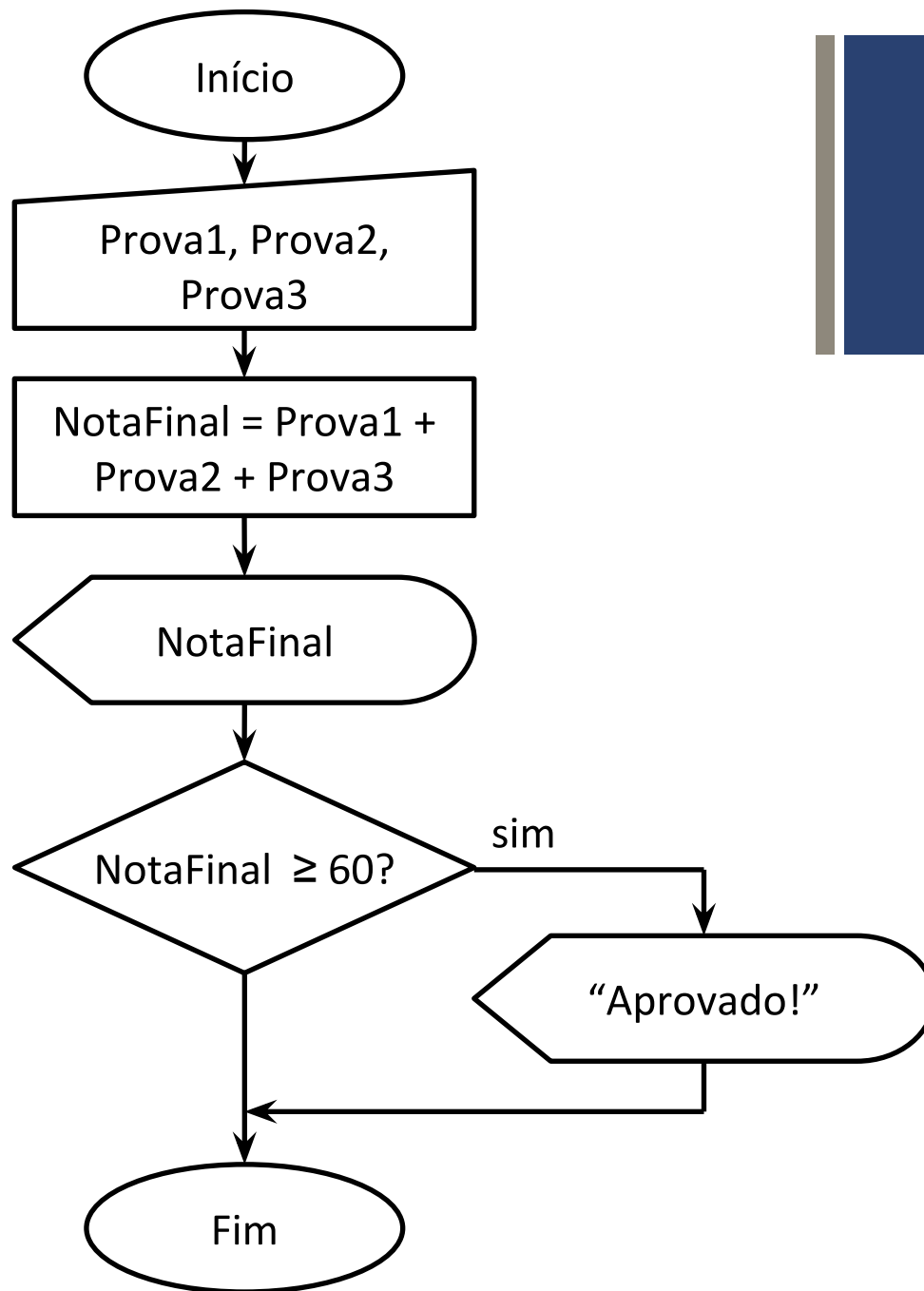




Na aula anterior...

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5
6     int prova1, prova2, prova3, notafinal;
7
8     //Ler notas da prova1, prova2, prova3
9     cout << "Digite sua nota na prova 1: ";
10    cin >> prova1;
11    cout << "Digite sua nota na prova 2: ";
12    cin >> prova2;
13    cout << "Digite sua nota na prova 3: ";
14    cin >> prova3;
15
16    //Calcular a nota final: somar as 3 notas
17    notafinal = prova1 + prova2 + prova3;
18
19    //Escrever a nota final
20    cout << "Sua nota final foi ";
21    cout << notafinal << endl;
22
23    if (notafinal >= 60)
24        cout << "Aprovado!" << endl;
25
26    return 0;
27 }
```

+ Na aula
anterior...



+

Na aula
anterior...

```
#include <iostream>
using namespace std;

int main() {

    int prova1, prova2, prova3, notafinal;

    //Ler notas da prova1, prova2, prova3
    cout << "Digite sua nota na prova 1: ";
    cin >> prova1;
    cout << "Digite sua nota na prova 2: ";
    cin >> prova2;
    cout << "Digite sua nota na prova 3: ";
    cin >> prova3;

    //Calcular a nota final: somar as 3 notas
    notafinal = prova1 + prova2 + prova3;

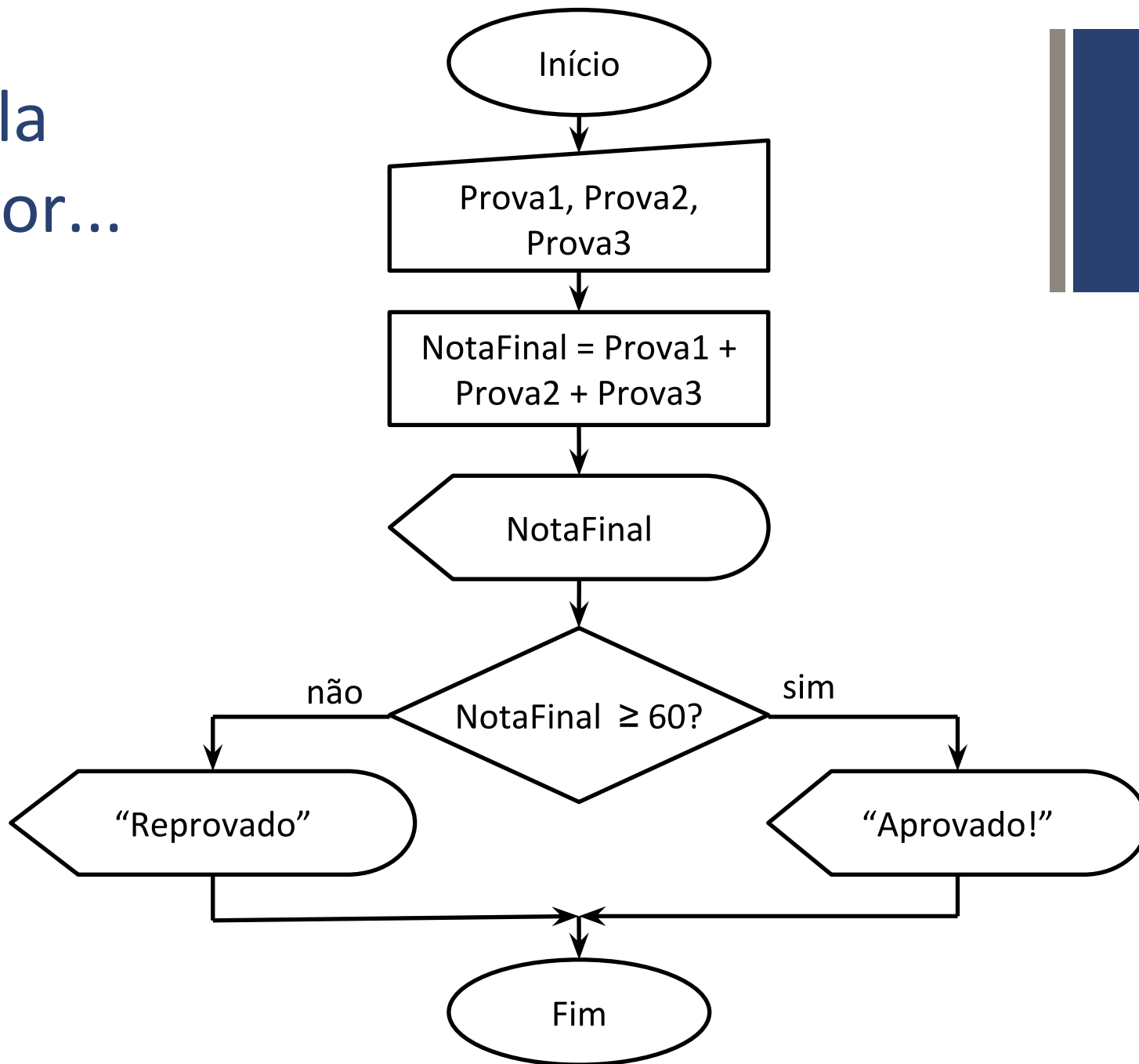
    //Escrever a nota final
    cout << "Sua nota final foi ";
    cout << notafinal << endl;

    if (notafinal >= 60)
        cout << "Aprovado!" << endl;
    else
        cout << "Reprovado." << endl;

    return 0;
}
```

+

Na aula
anterior...



+ Completando o programa

- Precisamos agora informar se o estudante foi aprovado, reprovado ou ficou de prova final
 - Para ser aprovado, deve ter nota no mínimo 60
 - Para ter direito à prova final, deve ter nota no mínimo 40
 - Com nota abaixo de 40 está reprovado sem direito a prova final
- Resumindo:
 - $\text{Nota} \geq 60$ → Aprovado
 - $40 \leq \text{Nota} < 60$ → Prova final
 - $\text{Nota} < 40$ → Reprovado

+ Código

- Versão somente com `if` (sem `else`)

```
if (notafinal >= 60)
    cout << "Aprovado" << endl;
if (notafinal >= 40 && notafinal < 60)
    cout << "Prova final" << endl;
if (notafinal < 40)
    cout << "Reprovado" << endl;
```

+ Código

- Versão somente com `if` (sem `else`)

```
if (notafinal >= 60)
    cout << "Aprovado" << endl;
if (notafinal >= 40 && notafinal < 60)
    cout << "Prova final" << endl;
if (notafinal < 40)
    cout << "Reprovado" << endl;
```

- Note que em C/C++ não está correto fazer `if (40 <= notafinal < 60)` para verificar se `notafinal` está no intervalo `[40, 60)`.

+ Código

- Versão somente com `if` (sem `else`)

```
if (notafinal >= 60)
    cout << "Aprovado" << endl;
if (notafinal >= 40 && notafinal < 60)
    cout << "Prova final" << endl;
if (notafinal < 40)
    cout << "Reprovado" << endl;
```

- Note que em C/C++ não está correto fazer `if (40 <= notafinal < 60)` para verificar se `notafinal` está no intervalo [40, 60).
- Isso compararia `notafinal` com 40, e o resultado da comparação (verdadeiro ou falso) seria comparado com 60. Totalmente sem sentido.

+ Código

- Versão com if-else “em cascata” (um dentro do outro)

```
if(notafinal >= 60)
    cout << "Aprovado" << endl;
else if (notafinal >= 40)
    cout << "Prova final" << endl;
else
    cout << "Reprovado" << endl;
```

- O(s) comando(s) do `if` ou `else` podem ser quaisquer comandos, inclusive outro `if-else`...
- Note que o segundo `if` não verifica se a `notafinal` é menor que 60. Por quê?

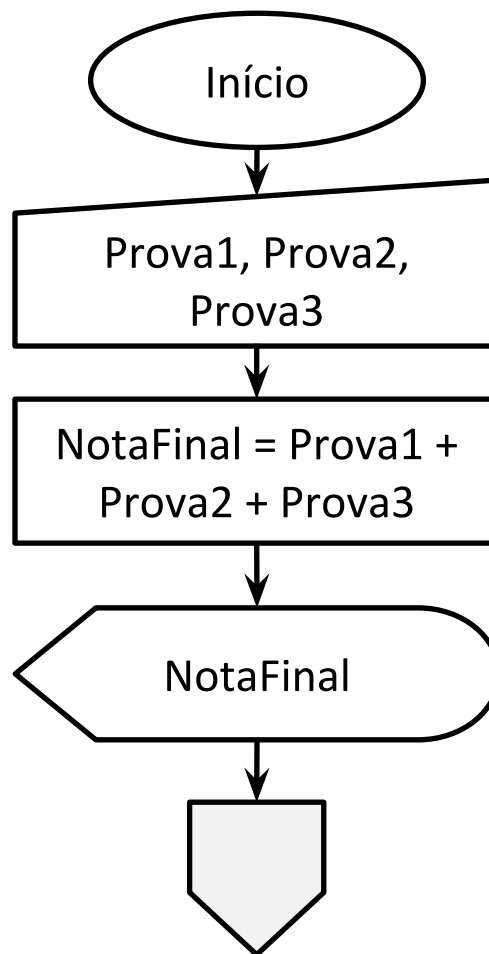
+ Código

- Versão com if-else “em cascata” (um dentro do outro)

```
if(notafinal >= 60)
    cout << "Aprovado" << endl;
else if (notafinal >= 40)
    cout << "Prova final" << endl;
else
    cout << "Reprovado" << endl;
```

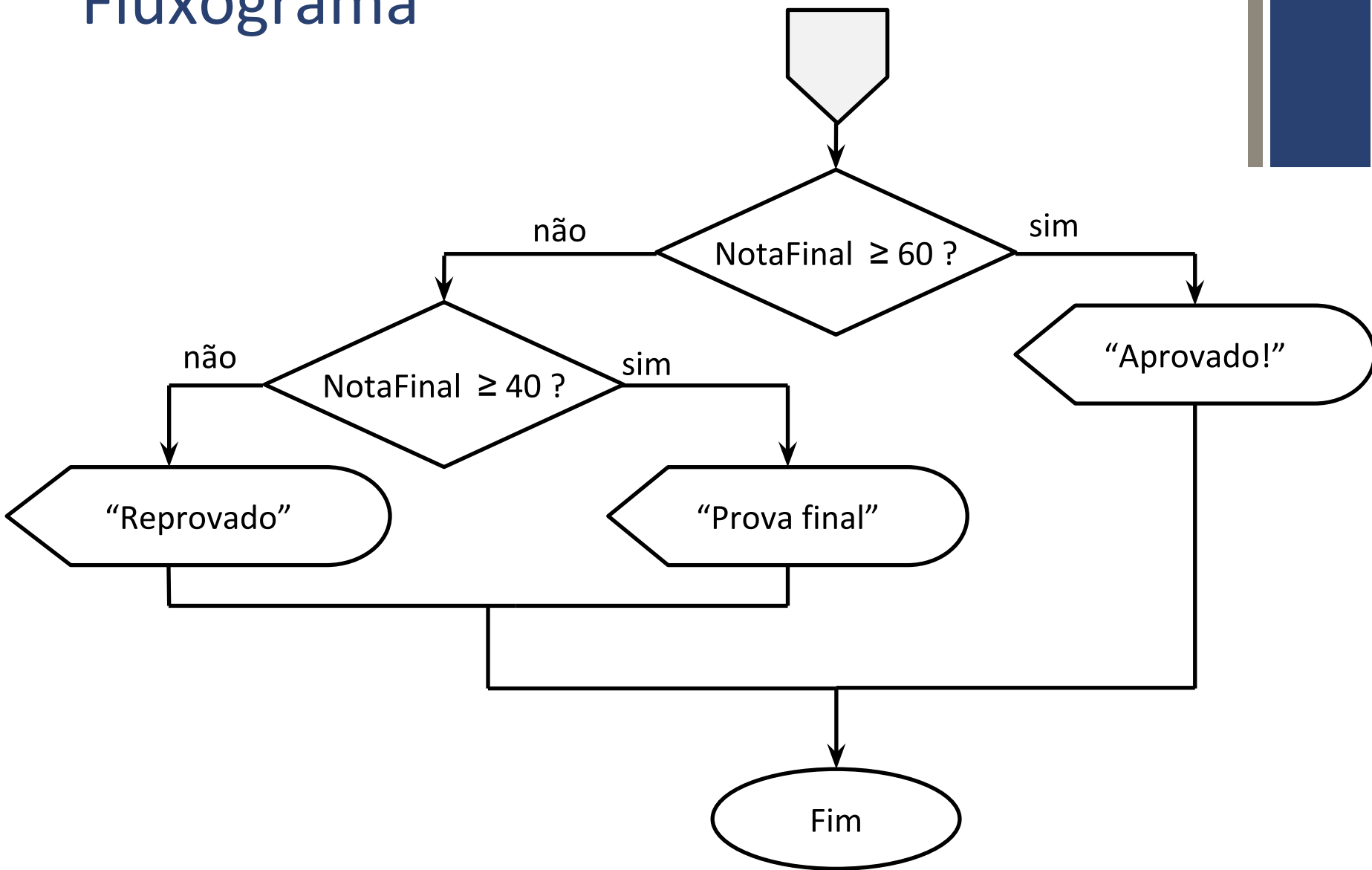
- O(s) comando(s) do `if` ou `else` podem ser quaisquer comandos, inclusive outro `if-else`...
- Note que o segundo `if` não verifica se a `notafinal` é menor que 60. Se chegou ali é porque é menor, do contrário teria escrito “Aprovado”

+ Fluxograma





Fluxograma





Exercício #1



- Ler a altura (em m) e a massa (em Kg) de uma pessoa, calcular seu IMC (Índice de Massa Corporal) e determinar sua classificação no IMC
- O IMC é a divisão da massa pelo quadrado da altura
- A classificação é dada de acordo com a tabela

IMC	Classificação
abaixo de 18.5	Abaixo do peso
de 18.5 a < 25	Peso normal
de 25 a < 30	Acima do peso
a partir de 30	Obesidade



Exercício #2



- Fazer um programa para calcular o preço final de um produto.
 - O vendedor informa o preço normal do produto
 - O programa pergunta se tem algum desconto
 - O vendedor respondem 'S' (sim) ou 'N' (não)
 - Se a resposta for 'S', o programa pergunta a porcentagem (%) de desconto
 - O programa informa o preço final



Exercício #3



- Ler uma expressão com dois operandos e informar o resultado
- A expressão será da forma **A operador B**
 - **A** e **B** são dois números inteiros
 - **operador** é um dentre os caracteres **+ - * /**
- Exemplo:
 - Entrada: 3 + 5 Saída: 8
 - Entrada: 5 * 8 Saída: 40

+ “Dangling else” (ambiguidade if-else)

- Ambiguidade que surge em `else` colocado em `if`'s aninhados
- Exemplo
 - Um empresa de e-commerce de MG cobra R\$30 pelo frete
 - Se a entrega é no próprio estado, o frete cai pra R\$15
 - Para compras a partir de R\$100 o frete é gratuito, independente do local de entrega

+

Código errado!

```
#include <iostream>
using namespace std;

int main()
{
    int valorcompra;
    int frete;
    char emMinas;

    cout << "Valor da compra: ";
    cin >> valorcompra;

    cout << "Entrega em Minas Gerais? (s/n): ";
    cin >> emMinas;

    frete = 30; //Valor padrao

    if (valorcompra < 100) //Tem que pagar frete
        if (emMinas == 's')
            frete = 15;
    else
        frete = 0;

    cout << "Valor do frete: " << frete << endl;

    return 0;
}
```

+ “Dangling else” (ambiguidade if-else)

- O `else` é de qual `if` ?


```
if (condicao1)
    if (condicao2)
        ...
    else
        ...
```

```
if (condicao1)
    if (condicao2)
        ...
else
    ...
```

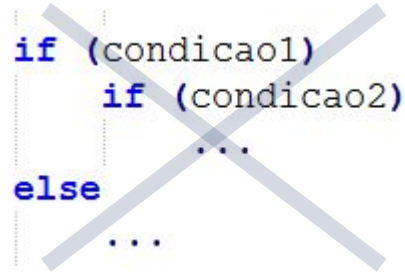
+ “Dangling else” (ambiguidade if-else)

- O `else` é de qual `if` ?

```
if (condicao1)
  if (condicao2)
    ...
  else
    ...
```



```
if (condicao1)
  if (condicao2)
    ...
else
  ...
```




- Para não haver ambiguidade, ele é sempre ligado ao `if` anterior

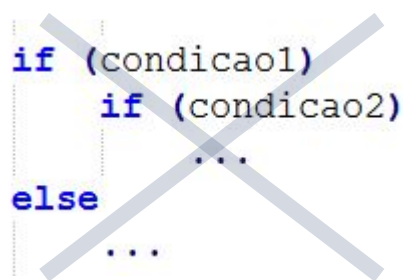
+ “Dangling else” (ambiguidade if-else)

- O `else` é de qual `if` ?


```
if (condicao1)
  if (condicao2)
    ...
  else
    ...
```



```
if (condicao1)
  if (condicao2)
    ...
else
  ...
```



```
if (condicao1)
{
  if (condicao2)
    ...
}
else
  ...
```



- Para não haver ambiguidade, ele é sempre ligado ao `if` anterior
- Se quiser que seja de outra forma, usar `{ }` para delimitar bloco

+ Código

- Note as { } para deixar o else para o if externo

```
#include <iostream>
using namespace std;

int main()
{
    int valorcompra;
    int frete;
    char emMinas;

    cout << "Valor da compra: ";
    cin >> valorcompra;

    cout << "Entrega em Minas Gerais? (s/n): ";
    cin >> emMinas;

    frete = 30;           //Valor padrao

    if (valorcompra < 100) //Se <100, tem que pagar frete
    {
        if (emMinas == 's') //Se e' de Minas
            frete = 15;     //altera para valor mais barato
    }
    else                  //Senao (se mais que R$100)
        frete = 0;        //altera para frete 0

    cout << "Valor do frete: " << frete << endl;

    return 0;
}
```

+ Operador condicional



- Existe um operador condicional em C/C++
 - É um operador ternário (o único da linguagem)
 - O resultado é a avaliação de uma entre duas expressões
- Sintaxe:
 - `<condição> ? <expressãoV> : <expressãoF>`
- Funcionamento
 - A condição (primeiro operando) é avaliada
 - Se for verdadeira, o resultado é a `expressãoV` (segundo operando)
 - Se for falsa, o resultado é a `expressãoF` (terceiro operando)

+ Op. ternário – Exemplo #1

- Ler dois números inteiros e informar o maior

```
#include <iostream>
using namespace std;
```

```
int main() {
    int a, b;
    int maior;

    cin >> a >> b;

    if (a>b)
        maior = a;
    else
        maior = b;

    cout << maior;

    return 0;
}
```

```
#include <iostream>
using namespace std;
```

```
int main() {
    int a, b;
    int maior;

    cin >> a >> b;

    maior = a>b?a:b;

    cout << maior << endl;

    return 0;
}
```

·sões que guardam o maior, depois imprimem)

+ Op. ternário – Exemplo #1

- Ler dois números inteiros e informar o maior

```
#include <iostream>
using namespace std;

int main()
{
    int a, b;

    cin >> a >> b;

    if (a>b)
        cout << a << endl;
    else
        cout << b << endl;

    return 0;
}
```

```
#include <iostream>
using namespace std;

int main()
{
    int a, b;

    cin >> a >> b;

    cout << (a>b?a:b) << endl;

    return 0;
}
```

(versoes que imprimem diretamente)

+ Op. ternário – Exemplo #2

- Ler um número inteiro e informar se ele é par ou ímpar

```
#include <iostream>
using namespace std;

int main()
{
    int x;

    cin >> x;

    cout << x;
    if (x%2 == 0)
        cout << " e' par.";
    else
        cout << " e' impar.";
    cout << endl;

    return 0;
}
```

```
#include <iostream>
using namespace std;

int main()
{
    int x;

    cin >> x;

    cout << x;
    cout << (x%2==0?" e' par.":" e' impar.");
    cout << endl;

    return 0;
}
```

+ Operador condicional



- Vantagem
 - Pode ser usado dentro de uma expressão! (if-else não)
- Desvantagem
 - Não é muito legível
 - Use com moderação!