

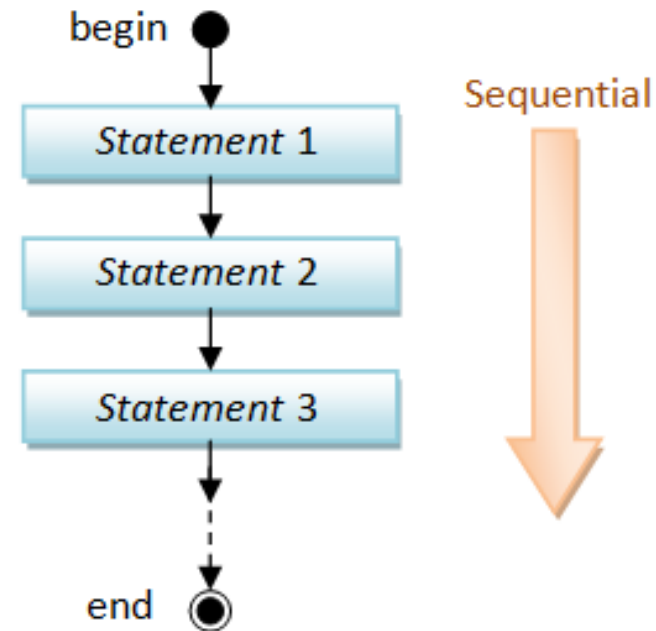
Introdução à Programação C++ para iniciantes e programadores de primeira viagem – Parte 2

Uberlândia, 22 de outubro de 2014

Daniel Teodoro Gonçalves Mariano
dtgmariano@gmail.com

O que é um programa ?

Um programa é uma sequência de instruções (chamadas instruções de programação) executada uma após a outra. Acontece de forma sequencial geralmente.



O que é uma variável?

Programas de computador manipulam (ou processam) informações (data).

Uma variável é usada para armazenar um pedaço de informação para processamento.

É chamada de variável pois seu valor armazenado pode ser alterado.

NAME	VALUE	TYPE
number	123	int
sum	-456	int
pi	3.1416	double
average	-55.66	double

Uma variável possui um **nome**, armazena um **valor** de um **tipo** declarado

Revisão: Estrutura básica de um programa em C++

```
- //Programa básico em C++  
  L #include <iostream>  
    using namespace std;  
- int main()  
  {  
    cout<<"Hello World!"<<endl;  
    system("PAUSE");  
    return 0;  
  }
```

Revisão: Cout e Cin

```
//Programa básico em C++

#include <iostream>

using namespace std;

int main()
{
    int num;

    cout<<"Entre um numero inteiro qualquer"<<endl;
    cin>>num;
    cout<<"O numero digitado foi: "<<num<<endl;

    system("pause");

    return 0;
}
```

Exercício

- Escreva um programa que faça as seguintes ações:
 - Armazene o somatório dos números contidos no intervalo [1, 2, 3, ..., 8, 9, 10] em uma variável chamada “soma”;
 - Imprima o resultado no console;

$$\text{soma} = \sum_{i=1}^{10} i$$

```
cout<<"O somatório do intervalo é "<<soma<<endl;
```

10. E se você precisar somar 1000 números?

E se você precisar escrever um programa que necessite somar 1000 números?

Como vocês implementariam?

Você estaria disposto a escrever mais de 1000 linhas de código?

Escrever 1000 linhas de código é uma opção, mas ela é a melhor forma?

Problema

Escreva um programa que some todos os números inteiros de 1 até n (n é um limite superior fornecido pelo usuário)

Existe alguma estrutura de repetição (LOOP) ?

Pode-se utilizar um loop do tipo **while**

```
1  /*
2   * Sum from 1 to an upperbound using a while-loop (SumNumbers.cpp).
3   */
4  #include <iostream>
5  using namespace std;
6
7  int main() {
8      int sum = 0;    // Declare an int variable sum to accumulate the numbers
9                      // Set the initial sum to 0
10     int upperbound; // Sum from 1 to this upperbound
11
12     // Prompt user for an upperbound
13     cout << "Enter the upperbound: ";
14     cin >> upperbound;
15
16     // Use a loop to repeatedly add 1, 2, 3,..., up to upperbound
17     int number = 1;
18     while (number <= upperbound) {
19         sum = sum + number; // accumulate number into sum
20         ++number;          // increment number by 1
21     }
22     // Print the result
23     cout << "The sum from 1 to " << upperbound << " is " << sum << endl;
24
25     return 0;
26 }
```


Vamos analisar o problema

Declara-se uma variável do tipo **int** nomeada de **sum** e inicializa a mesma com valor **0**

Essa variável será utilizada para acumular os números sobre as etapas do ciclo repetitivo

O usuário recebe a mensagem pedindo que insira um número para limite superior

Loop While para realizar o somatório dos números de 1 a upperbound.

Imprime o resultado

Fim do programa!

```
1  /*
2   * Sum from 1 to an upperbound using a while-loop (SumNumbers.cpp).
3   */
4  #include <iostream>
5  using namespace std;
6
7  int main() {
8      int sum = 0;    // Declare an int variable sum to accumulate the numbers
9                      // Set the initial sum to 0
10     int upperbound; // Sum from 1 to this upperbound
11
12     // Prompt user for an upperbound
13     cout << "Enter the upperbound: ";
14     cin >> upperbound;
15
16     // Use a loop to repeatedly add 1, 2, 3,..., up to upperbound
17     int number = 1;
18     while (number <= upperbound) {
19         sum = sum + number; // accumulate number into sum
20         ++number;           // increment number by 1
21     }
22     // Print the result
23     cout << "The sum from 1 to " << upperbound << " is " << sum << endl;
24
25     return 0;
26 }
```

O que é um teste lógico?

```
initialization-statement;  
while (test) {  
    loop-body;  
}  
next-statement;
```

Operadores de comparação

Operador	Significado	Exemplo
==	Igual a	x == y
!=	Diferente de	x != y
>	Maior que	x > y
>=	Maior ou igual a	x >= y
<	Menor que	x < y
<=	Menor ou igual a	x <= y

Exercício

num1	operador	num2
5	==	4.0
21	!=	7
3	>	2.99
9.0	>=	9
5	<	3.795
3	<=	3.015
3.1475	!=	3.1475
7	<	6.9999
3.000	==	3

resultado
False
True
True
True
False
True
False
False
true

Operadores de comparação

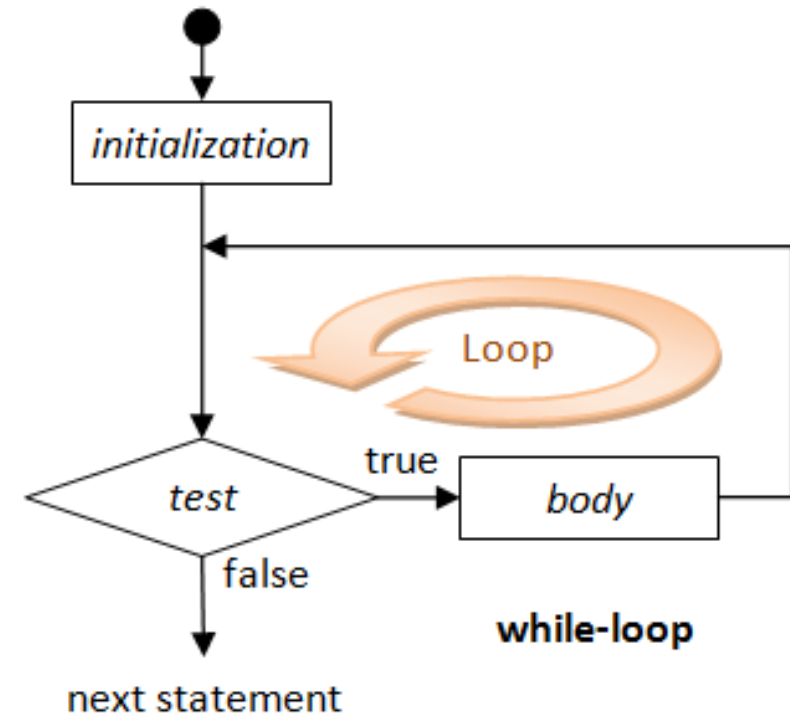
Operador	Significado	Exemplo
&&	E	(x >= 1) && (x <= 100)
	Ou	(x < 1) (x > 100)
!	Não	!(x == 8)

Exercício

num1	operador	num2	Resultado
True		True	True
True		False	True
False		True	True
False		False	False
True	&&	True	True
True	&&	False	False
False	&&	True	False
False	&&	False	False
True	!		False
False	!		True

Como funciona o while-loop?

```
initialization-statement;  
while (test) {  
    Loop-body;  
}  
next-statement;
```



Vamos praticar?

Faça as seguintes modificações no programa anterior:

- 1) Some todos os números entre um **lowerbound** e o **upperbound** fornecidos pelo usuário;
- 2) Some apenas os **números ímpares** entre **1** e um **upperbound** fornecido pelo usuário;
- 3) Some todos os números entre **1** e um **upperbound** que sejam **divisíveis por 7**. (Dica: Utilize "number = number + 7");
- 4) Some o quadrado de todos os números de 1 até o **upperbound**. (Dica: $1*1 + 2*2 + 3*3 + \dots$);

Problemas 2 e 3

Problema 2:

Crie um programa que compute o **produto** de todos os números de 1 a 10.

(Dica: Use uma variável chamada **product** no lugar de **sum** e inicialize ela com 1.)

Problema 3:

Crie um programa que escreve o fatorial de **n**, onde **n** é um valor inteiro entre 1 a 12.

Desafio!!!

Crie um programa que escreva os n primeiros termos da sequência de **Fibonacci** e o somatório dos mesmos

A sequência de Fibonacci é dada pela fórmula:

$$F(n) = \begin{cases} 0, & \text{se } n = 0; \\ 1, & \text{se } n = 1; \\ F(n-1) + F(n-2) & \text{outros casos.} \end{cases}$$

Resolução desafio

```
#include <iostream>

using namespace std;

int main()
{
    int n, num1, num2, temp, soma;    //Declarando variáveis

    cin>>n;                          //Número de elementos

    num1 = 0;                        //Inicializa o num1 com valor 0 pois F0 = 0
    num2 = 1;                        //Inicializa o num1 com valor 1 pois F1 = 1
    soma = num1;                     //Inicializa a soma com o valor de F0

    cout<<num1<<" ";               //Imprimi o primeiro elemento

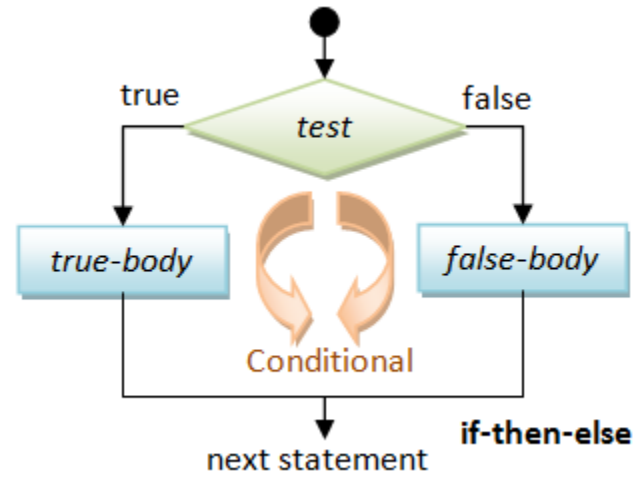
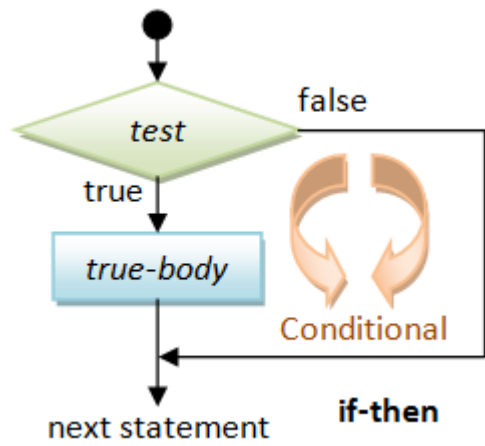
    for(int i=0; i<(n-1); i++)
    {
        cout<<num2<<" ";           //Imprimi o proximo elemento
        soma = soma + num2;         //Adiciona o proximo num (num2)
        temp = num2 + num1;         //cria uma var temp que é proximo elemento da seq. fib.
        num1 = num2;                //n2 para n1
        num2 = temp;                //n3 para n2
    }

    cout<<"\nsoma = "<<soma<<endl;  //Imprimi soma
    system("pause");
    return 0;                       //Fim programa!
}
```

11. Condicional (ou decisão)

- Suponha que você criou uma variável que armazena um número X qualquer.
- Se lhe for perguntado se esse número é maior que Y como imprimir respostas para as diferentes possibilidades?
- Possibilidades
 - $X > Y$
 - $X == Y$
 - $X < Y$
- Existe alguma estrutura condicional que resolva esse problema?

If e If-else



Exemplo

```
// Return true if x is between 0 and 100 (inclusive)
(x >= 0) && (x <= 100) // AND (&&)
// Incorrect to use 0 <= x <= 100

// Return true if x is outside 0 and 100 (inclusive)
(x < 0) || (x > 100) // OR (||)
!((x >= 0) && (x <= 100)) // NOT (!), AND (&&)

// Return true if "year" is a leap year
// A year is a leap year if it is divisible by 4 but not by 100, or it is divisible by 400.
((year % 4 == 0) && (year % 100 != 0)) || (year % 400 == 0)
```

Problema: Notas dos estudantes

Um professor está com dificuldade em fechar as notas do semestre, por isso ele pediu ajuda a você para ajudá-lo no seguinte problema:

Ele precisa classificar as notas dos alunos em conceitos A, B, C, D, E e F de acordo com a tabela ao lado:

Crie um programa que receba um número inteiro referente a nota final de um estudante e imprima a classificação dessa nota!

Nota	
A	10
B	$10 > N \geq 8$
C	$8 > N \geq 6$
D	$6 > N \geq 4$
E	$4 > N \geq 2$
F	$2 > N \geq 0$

Tipo double e números de ponto flutuante

Uma variável possui um nome e um tipo e pode armazenar valores desse tipo particular apenas.

Até o momento utilizamos variáveis do tipo int

Uma variável int armazena apenas valores inteiros

Em programação números reais como 3.1416 e -55.66 são chamados de ponto flutuante e pertencem a um tipo de variável chamado double

Problema: Média de um aluno

- Escreva um programa que receba de um aluno as notas das 4 provas da matéria de Cálculo I.
- Calcule a média.
- Imprima o resultado.

```

1  /*
2   * Convert temperature between Celsius and Fahrenheit
3   * (ConvertTemperature.cpp)
4   */
5  #include <iostream>
6  using namespace std;
7
8  int main() {
9      double celsius, fahrenheit;
10
11     cout << "Enter the temperature in celsius: ";
12     cin >> celsius;
13     fahrenheit = celsius * 9.0 / 5.0 + 32.0;
14     cout << celsius << " degree C is " << fahrenheit << " degree F." << endl << endl;
15
16     cout << "Enter the temperature in fahrenheit: ";
17     cin >> fahrenheit;
18     celsius = (fahrenheit - 32.0) * 5.0 / 9.0;
19     cout << fahrenheit << " degree F is " << celsius << " degree C." << endl;
20
21     return 0;
22 }

```

```

Enter the temperature in celsius: 37.2
37.2 degree C is 98.96 degree F.

```

```

Enter the temperature in fahrenheit: 100
100 degree F is 37.7778 degree C.

```

Problema: Cálculo IMC

O IMC (Índice de Massa Corporal) é um cálculo simples e pode ajudar a determinar se uma pessoa está em seu peso ideal ou não.

Dado a altura (**H**) em metros e o peso (**P**) em kilos, determine o IMC e classifique de acordo com a tabela a seguir:

IMC	Classificação
< 18,5	Abaixo do peso
18,6 – 24,9	Saudável
25 – 29,9	Sobrepeso
30,0 – 34,9	Obesidade Grau I
35,0 – 39,9	Obesidade Grau II
>= 40	Obesidade Grau III

Crie um programa que seja capaz de receber a altura e o peso do usuário, calcule o seu IMC e imprima a classificação do mesmo.

$$IMC = \frac{P}{A^2}$$