



LÓGICA DE PROGRAMAÇÃO PARA ENGENHARIA INTRODUÇÃO À PROGRAMAÇÃO COM C/C++

Prof. Dr. Daniel Caetano

2011 - 2

Visão Geral

1

- Transição Portugal → C/C++

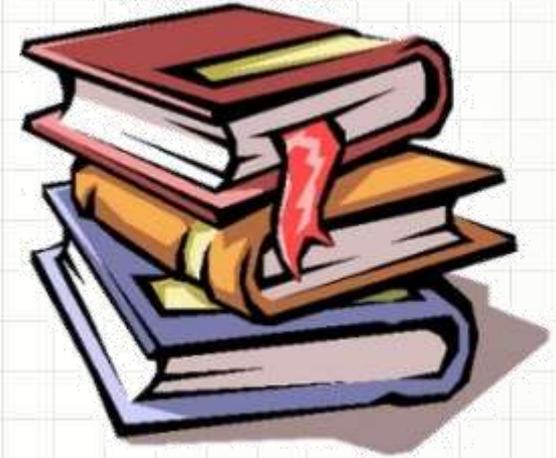
2

- A Sintaxe do C/C++

3

- Construindo um Programa

Material de Estudo



Material

Acesso ao Material

Notas de Aula

<http://www.caetano.eng.br/aulas/lpe/>
(Aula 5)

Apresentação

<http://www.caetano.eng.br/aulas/lpe/>
(Aula 5)

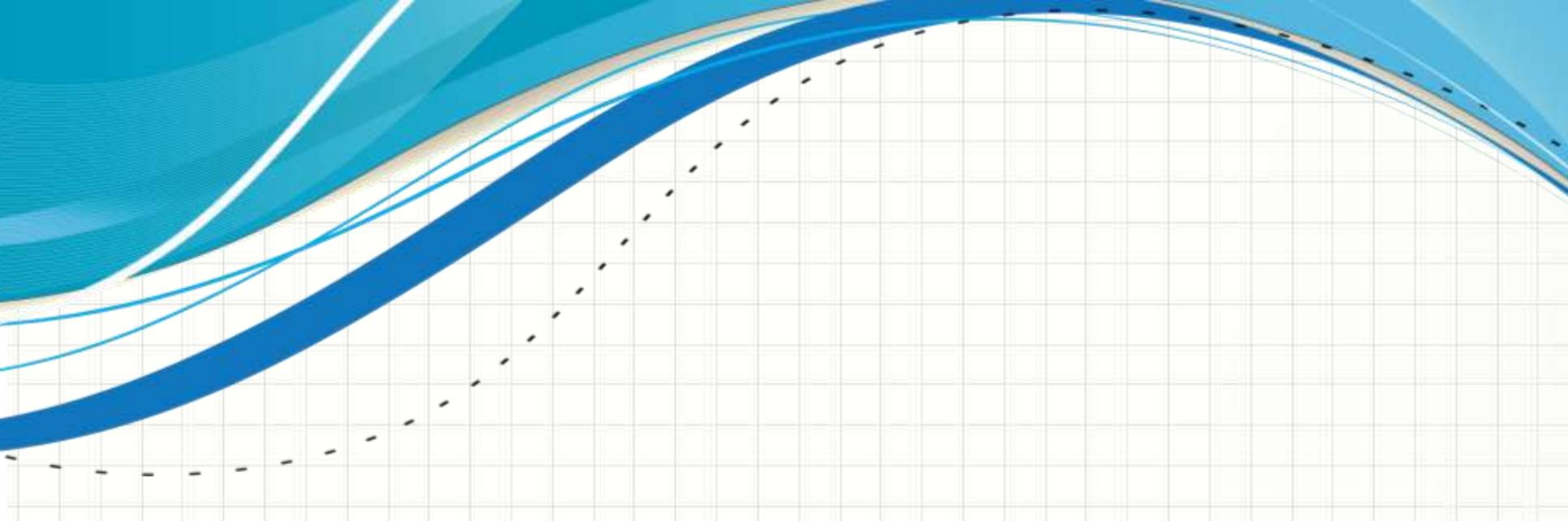
Material Didático

Lógica de Programação – Fundamentos da
Programação de Computadores, páginas 7 a 49.

Objetivos

- Entender o mecanismo de um programa em C/C++
- Apresentar a estrutura da Linguagem C/C++
- Capacitar o aluno para escrever algoritmos sequenciais simples em C/C++
- **LISTA 1**





MIGRANDO PARA O C/C++

Por que C/C++?

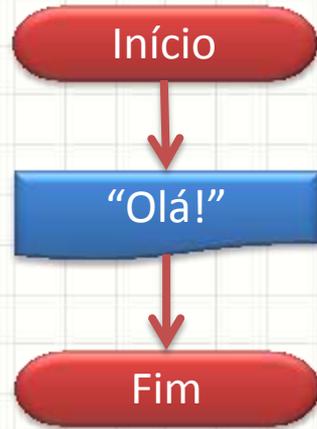
- Português Estruturado: Limitado
- Programas mais úteis: C/C++
 - Ver o resultado!
- Detalhamento:
 - Linguagem natural
 - Fluxograma
 - Português estruturado
 - Linguagem de Programação (C/C++)



**Detalhamento
Aumenta**

Algoritmo Mais Simples do Mundo

- Fluxograma



- Portugol

Algoritmo “Imprime Olá”

Var

Inicio

Escreva (“Olá! “)

FimAlgoritmo

- Linguagem C

```
#include <stdio>
#include <iostream>
using namespace std;
int main(void)
{
    cout << “Olá!”;
    getch();
}
```

Algoritmo Mais Simples do Mundo

- Portugol

Algoritmo "Imprime Olá"

Var

Inicio

Escreva ("Olá! ")

FimAlgoritmo

- Linguagem C

```
#include <stdio>
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main(void)
```

```
{
```

```
    cout << "Olá!";
```

```
    getch();
```

```
}
```

Área de Declarações

Algoritmo Mais Simples do Mundo

- Portugol

Algoritmo "Imprime Olá"

Var

Inicio

Escreva ("Olá! ")

FimAlgoritmo

- Linguagem C

```
#include <stdio>
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main(void)
```

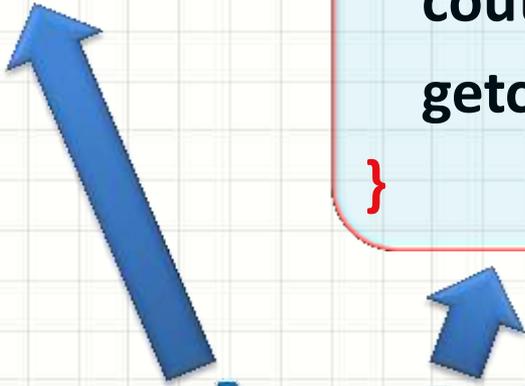
```
{
```

```
    cout << "Olá!";
```

```
    getch();
```

```
}
```

Área de Código



Algoritmo Mais Simples do Mundo

- Portugol

Algoritmo "Imprime Olá"

Var

Inicio

Escreva ("Olá! ")

FimAlgoritmo

- Linguagem C

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

```
int main(void)
```

```
{
    cout << "Olá!";
    getchar();
}
```

O que são?

Algoritmo Mais Simples do Mundo

- Portugol

Algoritmo "Imprime Olá"

Var

Início

Escreva ("Olá! ")

FimAlgoritmo

- Linguagem C

```
#include <stdio>
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main(void)
```

```
{
```

```
    cout << "Olá!";
```

```
    getch();
```

```
}
```

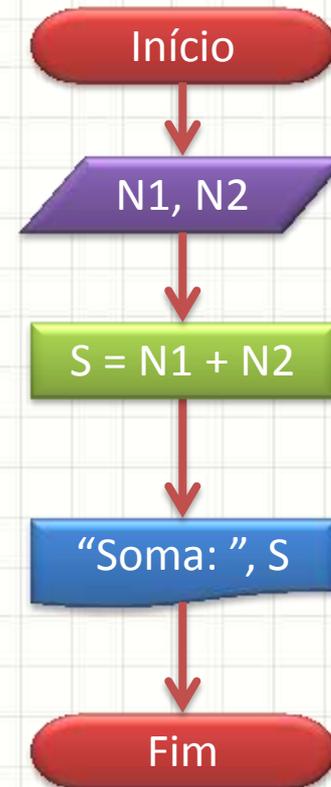
E as variáveis?

Somando Dois Números

- Linguagem Natural

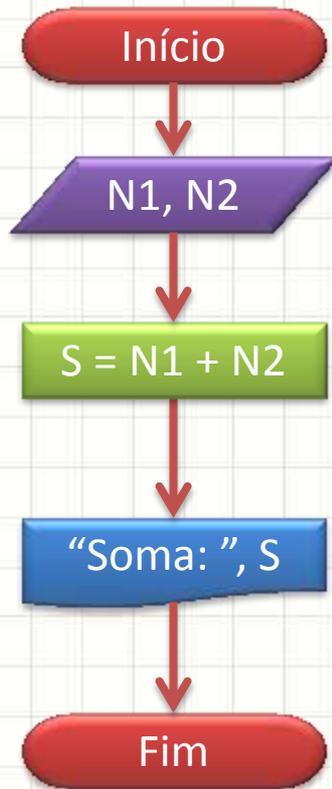
1. Leia dois números
2. Some os dois números
3. Imprima a soma

- Fluxograma



Somando Dois Números

- Fluxograma



- Portugol

Algoritmo “Soma 2 Números”

Var

N1, N2, S: INTEIRO

Início

Escreva(“Digite Um Número:”)

Leia(N1)

Escreva(“Digite Outro Número:”)

Leia(N2)

$S \leftarrow N1 + N2$

Escreva(“Soma:”, S)

FimAlgoritmo

Somando Dois Números

- Portugol

Algoritmo "Soma 2 Números"

Var

N1, N2, S: INTEIRO

Inicio

Escreva("Digite Um Número:")

Leia(N1)

Escreva("Digite Outro Número:")

Leia(N2)

$S \leftarrow N1 + N2$

Escreva("Soma:", S)

FimAlgoritmo

- Linguagem C

```
#include <stdio>
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main(void)
```

```
{
```

```
    int N1, N2, S;
```

```
    cout << "Digite Um Número:";
```

```
    cin >> N1;
```

```
    cout << "Digite Outro Número:";
```

```
    cin >> N2;
```

```
    S = N1 + N2;
```

```
    cout << "Soma: " << S;
```

```
    getchar();
```

```
}
```

Somando Dois Números

- Portugol

Algoritmo “Soma 2 Números”

Var

N1, N2, S: INTEIRO

Inicio

Escreva(“Digite Um Número:”)

Leia(N1)

Escreva(“Digite Outro Número:”)

Leia(N2)

$S \leftarrow N1 + N2$

Escreva(“Soma:”, S)

FimAlgoritmo

- Linguagem C

```
#include <stdio>
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main(void)
```

```
{
```

```
    int N1, N2, S;
```

```
    cout << “Digite Um Número:”;
```

```
    cin >> N1;
```

```
    cout << “Digite Outro Número:”;
```

```
    cin >> N2;
```

```
    S = N1 + N2;
```

```
    cout << “Soma: ” << S;
```

```
    getchar();
```

```
}
```

Somando Dois Números

- Portugol

Algoritmo "Soma 2 Números"

Var

N1, N2, S: INTEIRO

Início

Escreva("Digite Um Número:")

Leia(N1)

Escreva("Digite Outro Número:")

Leia(N2)

$S \leftarrow N1 + N2$

Escreva("Soma:", S)

Fim Algoritmo

- Linguagem C

```
#include <stdio>
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main(void)
```

```
{
```

```
int N1, N2, S;
```

```
cout << "Digite Um Número:";
```

```
cin >> N1;
```

```
cout << "Digite Outro Número:";
```

```
cin >> N2;
```

```
S = N1 + N2;
```

```
cout << "Soma:" << S;
```

Área de Variáveis

Calculando o Número de Semanas

- Linguagem Natural

1. Leia o número de dias
2. Divida o número de dias por 7 (dias por semana)
3. Imprima o número de semanas

- Fluxograma



Calculando o Número de Semanas

- Fluxograma



- Portugol

Algoritmo "Número de Semanas"

Var

DIAS, NS: **REAL**

Início

Escreva("Quantos dias? ")

Leia(DIAS)

$NS \leftarrow DIAS / 7$

Escreva("Semanas: ", NS)

FimAlgoritmo

Calculando o Número de Semanas

- Portugol

Algoritmo “Número de Semanas”

Var

DIAS, NS: **REAL**

Inicio

Escreva(“Quantos dias? ”)

Leia(DIAS)

$NS \leftarrow DIAS / 7$

Escreva(“Semanas: ”, NS)

FimAlgoritmo

- Linguagem C

```
#include <stdio>
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main(void)
```

```
{
```

```
    float DIAS, NS;
```

```
    cout << “Quantos dias? ”;
```

```
    cin >> DIAS;
```

```
    NS = DIAS / 7;
```

```
    cout << “Semanas: ” << NS;
```

```
    getchar();
```

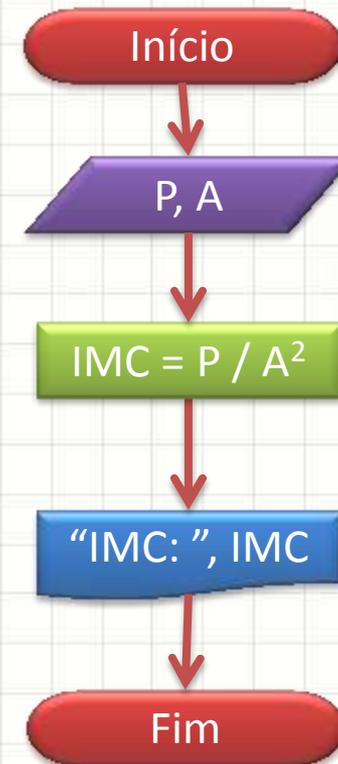
```
}
```

Índice de Massa Corporal

- Linguagem Natural

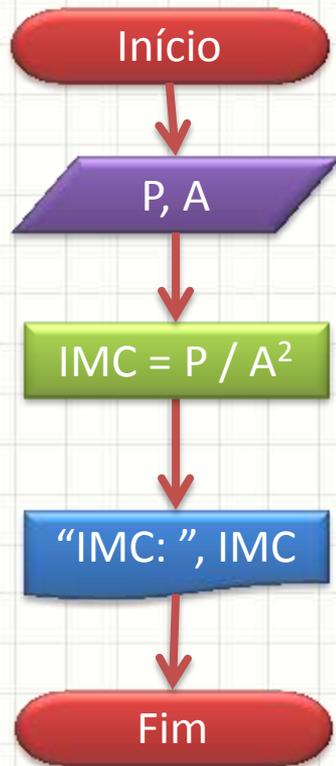
1. Leia o peso do sujeito, em kg
2. Leia a altura do sujeito, em metros
3. Calcule o IMC, dividindo o peso pelo quadrado da altura
4. Imprima o IMC

- Fluxograma



Índice de Massa Corporal

- Fluxograma



- Portugol

Algoritmo “Calcula IMC”

Var

P, A, IMC: **REAL**

Início

Escreva(“Digite o peso, em Kg: ”)

Leia(P)

Escreva(“Digite altura, em m: ”)

Leia(A)

$IMC \leftarrow P / (A * A)$

Escreva(“IMC: ”, IMC)

FimAlgoritmo

Índice de Massa Corporal

- Portugol

Algoritmo “Calcula IMC”

Var

P, A, IMC: **REAL**

Início

Escreva(“Digite o peso, em Kg: ”)

Leia(P)

Escreva(“Digite a altura, em m: ”)

Leia(A)

$IMC \leftarrow P / (A * A)$

Escreva(“IMC: ”, IMC)

FimAlgoritmo

- Linguagem C

```
#include <stdio>
```

```
#include <iostream>
```

```
using namespace std;
```

```
int main(void)
```

```
{
```

```
    float P, A, IMC;
```

```
    cout << “Digite o peso, em Kg: ”;
```

```
    cin >> P;
```

```
    cout << “Digite a altura, em m: ”;
```

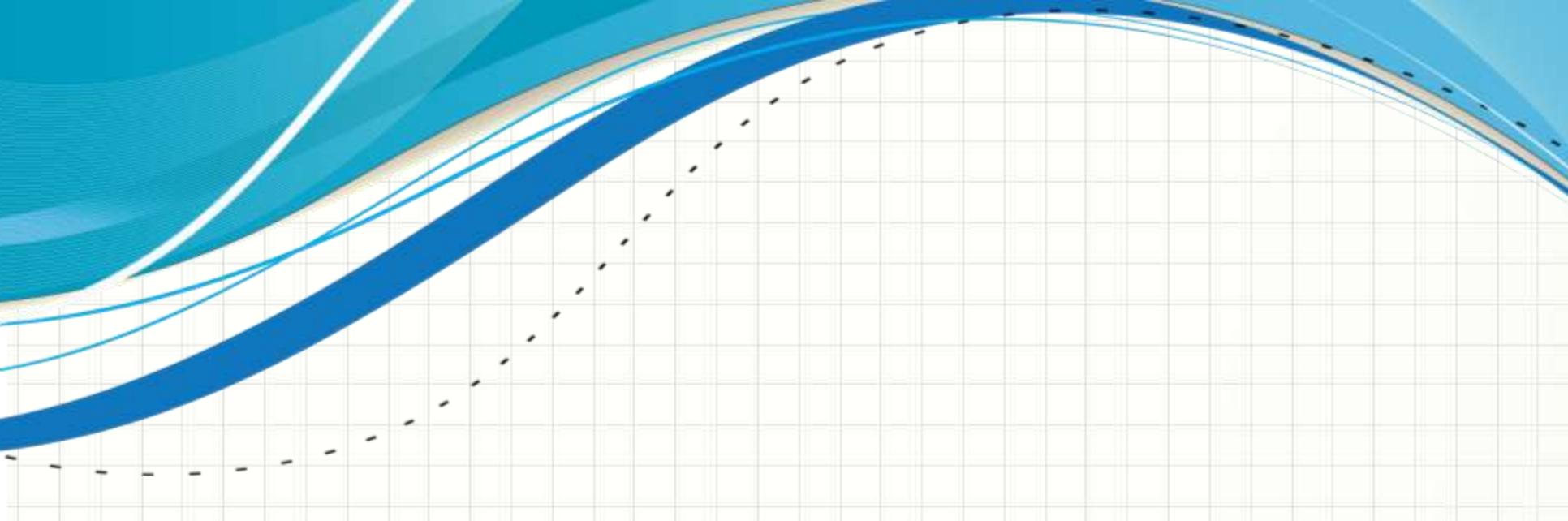
```
    cin >> A;
```

```
     $IMC = P / (A * A);$ 
```

```
    cout << “IMC: ” << IMC;
```

```
    getchar();
```

```
}
```



A LINGUAGEM C/C++

Estrutura do Programa

- Todo programa em C/C++ tem pelo menos **três** partes importantes:
 1. A Declaração
 2. O Código (Algoritmo)
 3. A área de variáveis
 - Fica na área de código
 - Primeira coisa!
- Há diferenciação entre maiúsculas e minúsculas!

```
#include <stdio>
#include <iostream>
using namespace std;
int main(void)
```

```
{
    cout << "Olá!";
    getch();
}
```

Estrutura do Programa

- Área de Declaração indica
 1. Bibliotecas usadas
 2. Nome da rotina
 - Sempre **main!**

```
#include <stdio>
#include <iostream>
using namespace std;
int main(void)
{
```

```
    cout << "Olá!";
    getch();
}
```

Estrutura do Programa

- Área de Código: Instruções ao programa
- Demarcada por { ... }
- Todas as linhas na área de código devem terminar com

;

```
#include <stdio>  
#include <iostream>  
using namespace std;  
int main(void)
```

```
{
```

```
    cout << "Olá!";  
    getch();
```

```
}
```

Estrutura do Programa

- Vamos declarar uma variável A para INTEIROS

```
#include <stdio>
#include <iostream>
using namespace std;
int main(void)
{
    int A;

    getchar();
}
```

A declaração de variáveis tem sempre o seguinte formato:

TipoDaVariável NomeDaVariável ;



```
#include <stdio.h>  
#include <iostream>  
using namespace std;  
int main(void)  
{  
    int A;
```

```
    getchar();
```

```
}
```

A declaração de variáveis tem sempre o seguinte formato:

TipoDaVariável NomeDaVariável ;

```
#include <stdio.h>
#include <iostream>
using namespace std;
int main(vo
```



int A;

```
getchar();
```

```
}
```

Observe que, em C/C++,
declaramos antes o TIPO e
depois o NOME

```
#include <stdio.h>  
#include <iostream>  
using namespace std;  
int main(void)  
{  
    int A;  
  
    getchar();  
}
```

Observe também que pelo
menos um espaço deve separar
o tipo do nome

```
#include <stdio.h>
#include <iostream>
using namespace std;
int main(void)
{
    int A;

    getch();
}
```

Estrutura do Programa

- Vamos declarar uma variável A para INTEIROS
- Declarando variável B também para INTEIROS

```
#include <stdio>
#include <iostream>
using namespace std;
int main(void)
{
    int A, B;

    getchar();
}
```

Podemos declarar mais de uma variável do mesmo tipo em uma única linha, separando-as por vírgulas

TipoDaVariável Variável1, Variável2 ;

```
#include <iostream>  
using namespace std;  
int main()  
{
```

```
int A, B;
```

```
getchar();
```

```
}
```

Estrutura do Programa

- Para declarar uma terceira variável C, para números reais....

```
#include <stdio>
#include <iostream>
using namespace std;
int main(void)
{
    int A, B;
    float C;

    getchar();
}
```

Para declarar variáveis de tipos diferentes, é necessário declarar em linhas diferentes

```
using namespace std;
```

```
int main()
```

```
{
```

```
int A, B;
```

```
float C;
```

```
getchar();
```

```
}
```

Estrutura do Programa

- Para declarar uma terceira variável C, para números reais....

```
#include <stdio>
#include <iostream>
using namespace std;
int main(void)
{
    int A, B;
    float C;

    getchar();
}
```

Simplificadamente, os tipos de variáveis são:

int (inteiros)

float (reais)

bool (lógicos)

char (caracteres)

Há regras para os NOMES das variáveis:

1. Não podem ter espaços
2. Não podem iniciar com números
3. Não podem usar caracteres especiais (acentos etc.)
4. Não podem ser uma palavra da linguagem (int ou main, por exemplo)

```
int main(void)
```

```
{
```

```
    int A, B;
```

```
    float C;
```

```
    getchar();
```

```
}
```

Área de Código

- Tarefa mais comum: escrever texto na tela
- Escrever “Uma frase qualquer”

```
#include <stdio>  
#include <iostream>  
using namespace std;  
int main(void)  
{
```

```
    cout << “Uma frase qualquer”;
```

```
    getchar();
```

```
}
```

Área

Uma frase qualquer

- Ta
- Es

la

```
cout << "Uma frase qualquer";
```

```
getchar();
```

```
}
```

Área de Código

- Escrever “Uma frase qualquer”
- E escrever o número **1934**

```
#include <stdio>
#include <iostream>
using namespace std;
int main(void)
{
```

```
    cout << “Uma frase qualquer”;
```

```
    cout << 1934;
```

```
    getchar();
```

```
}
```

Área

Uma frase qualquer1934

- Es
- E (

```
cout << "Uma frase qualquer";
```

```
cout << 1934;
```

```
getchar();
```

```
}
```

Área de Código

- Escrever “Uma frase qualquer”, **pular 1 linha**
- E escrever o número 1934

```
#include <stdio>
#include <iostream>
using namespace std;
int main(void)
{

    cout << “Uma frase qualquer”;
    cout << endl;
    cout << 1934;

    getch();
}
```

Área

Uma frase qualquer
1934

- Es
- E

nha



```
cout << "Uma frase qualquer";  
cout << endl;  
cout << 1934;  
getchar();  
}
```

Área de Código

- Escrever “2 + 3 =”
- E escrever o resultado de **2+3** ...

```
#include <stdio>
#include <iostream>
using namespace std;
int main(void)
{
```

```
    cout << “2 + 3 = ”;
    cout << 2+3;
```

```
    getch();
}
```

Área

$$2 + 3 = 5$$

- Es
- E

```
cout << "2 + 3 = ";
```

```
cout << 2+3;
```

```
getchar();
```

```
}
```

Área

2 + 3 = 5

- Es
- E

Quando mando escrever uma string (algo entre aspas, como "2+3="), o computador reproduz exatamente na tela

```
cout << "2 + 3 =";  
cout << 2+3;
```

```
getchar();  
}
```

Área

- Es
- E

2 + 3 = 5

Quando mando escrever uma expressão (algo sem **aspas**, como **2+3**), o computador **resolve** e **imprime o resultado**.

```
cout << "2 + 3 =";
```

```
cout << 2+3;
```

```
getchar();
```

```
}
```

Área de Código

- Se quiser que as impressões apareçam lado a lado, preciso usar duas linhas **cout** ?

```
#include <stdio>
#include <iostream>
using namespace std;
int main(void)
{
```

```
    cout << "2 + 3 = ";
    cout << 2+3;
```

```
    getch();
}
```

Área de Código

- **NÃO!** Podemos imprimir com uma linha só, separando as informações com o sinal <<

```
#include <stdio>
#include <iostream>
using namespace std;
int main(void)
{
```

```
    cout << "2 + 3 = " << 2+3 ;
```

```
    getch();
}
```

Áre

$$2 + 3 = 5$$

- N
se

só,
:

```
cout << "2 + 3 = " << 2+3 ;
```

```
getchar();  
}
```

Nas expressões, podemos usar as operações matemáticas usuais:

Soma:	+	Divisão:	/
Subtração:	-	Resto da Divisão:	%
Multiplicação:	*	Parênteses:	()

```
int main(void)  
{
```



```
cout << "2 + 3 = " << 2+3 ;
```

```
getchar();  
}
```

Área de Código

- Para fazer coisas mais elaboradas, usamos as variáveis

```
#include <stdio>
#include <iostream>
using namespace std;
int main(void)
{
    int SOMA;
    SOMA = 2 + 3;
    cout << "2 + 3 = " << SOMA;

    getchar();
}
```

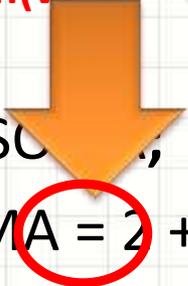
Área de Código

- Para fazer coisas mais interessantes com variáveis

O símbolo = (igual) é usado para atribuição; isso significa que ele serve para **guardar um valor em uma variável**

```
#include <stdio.h>
#include <iostream>
using namespace std;
int main(void)
{
    int SOMA;
    SOMA = 2 + 3;
    cout << "2 + 3 = " << SOMA;

    getch();
}
```



Uma atribuição sempre tem o nome de uma variável do lado esquerdo...

```
#include <iostream>
using namespace std;
int main(void)
{
    int SOMA;
    SOMA = 2 + 3;
    cout << "2 + 3 = " << SOMA;

    getchar();
}
```



...e um valor ou expressão
do lado direito

```
#include <iostream>
using namespace std;
int main(void)
{
    int SOMA,
    SOMA = 2 + 3;
    cout << "2 + 3 = " << SOMA;

    getch();
}
```



Como só é possível guardar números na memória, toda expressão é calculada para que seu resultado seja armazenado

```
#include <iostream>
using namespace std;
int main(void)
{
    int SOMA,
    SOMA = 2 + 3;
    cout << "2 + 3 = " << SOMA;

    getch();
}
```



Isso significa que o computador resolve a linha destacada assim:

SOMA = 2+3;

SOMA = 5;

```
#include <iostream>
using namespace std;
int main(void)
{
    int SOMA;
    SOMA = 2 + 3;
    cout << "2 + 3 = " << SOMA;

    getchar();
}
```



Área de Código

- Finalmente, falta vermos como ler um valor
- Note que precisamos de uma variável!

```
#include <stdio>
#include <iostream>
using namespace std;
int main(void)
{
    int A;
    cin >> A;

    getchar();
}
```

Área de Código

- Finalmente, falta ver
- Note que precisamos

```
#include <stdio.h>
#include <iostream>
using namespace std;
int main(void)
{
    int A;
    cin >> A;
    getchar();
}
```

O comando de entrada espera o usuário digitar algo e guarda na variável indicada (no caso, a variável A)

Área de Código

- Finalmente, falta ver
- Note que precisamos

```
#include <stdio.h>
#include <iostream>
using namespace std;
int main(void)
{
    int A;
    cin >> A;
    getchar();
}
```

Desta forma,
entretanto, nada
aparecerá na tela,
além de um cursor
(ou prompt)

Área -

- Fil
- No

da
ela,
tor

`cin >> A;`

`getchar();`

`}`

Área de Código

- Vamos imprimir uma mensagem para que o usuário saiba o que fazer!

```
#include <stdio>
#include <iostream>
using namespace std;
int main(void)
{
    int A;
    cout << "Digite um número: ";
    cin >> A;

    getchar();
}
```

Área

- Variáveis
usadas

Digite um número: _

e o

```
cout << "Digite um número: ";
```

```
cin >> A;
```

```
getchar();
```

```
}
```

Área de Código

- Vamos imprimir uma mensagem para o usuário saiba o que f

O tipo de informação que o computador irá aceitar, depende do tipo da variável!

```
#include <stdio.h>
#include <iostream>
using namespace std;
int main(void)
```

```
{
```

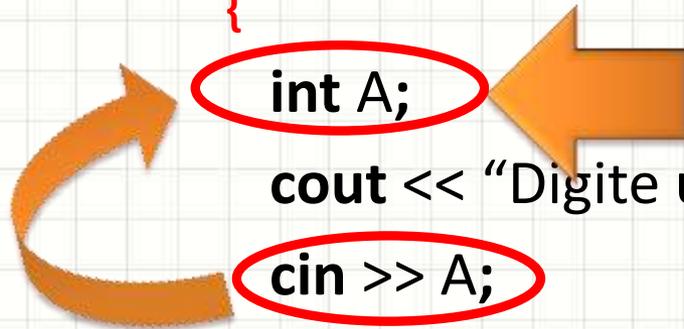
```
    int A;
```

```
    cout << "Digite um número: ";
```

```
    cin >> A;
```

```
    getch();
```

```
}
```



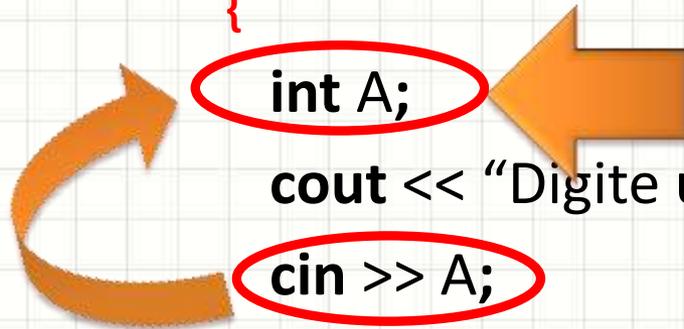
Área de Código

- Vamos imprimir uma mensagem para o usuário saiba o que fazer

```
#include <stdio.h>
#include <iostream>
using namespace std;
int main(void)
{
    int A;
    cout << "Digite um número: ";
    cin >> A;

    getch();
}
```

Se a variável é inteira e o usuário digitar qualquer coisa que não seja um inteiro, a linguagem C/C++ causará um erro



Área de Código

- Agora que já temos um valor em A, podemos imprimi-lo!

```
#include <stdio>
#include <iostream>
using namespace std;
int main(void)
{
    int A;
    cout << "Digite um número: ";
    cin >> A;
    cout << "O número digitado foi: " << A;
    getch();
}
```

Área

- Ag
im

Digite um número: 1357
O número digitado foi: 1357

mos

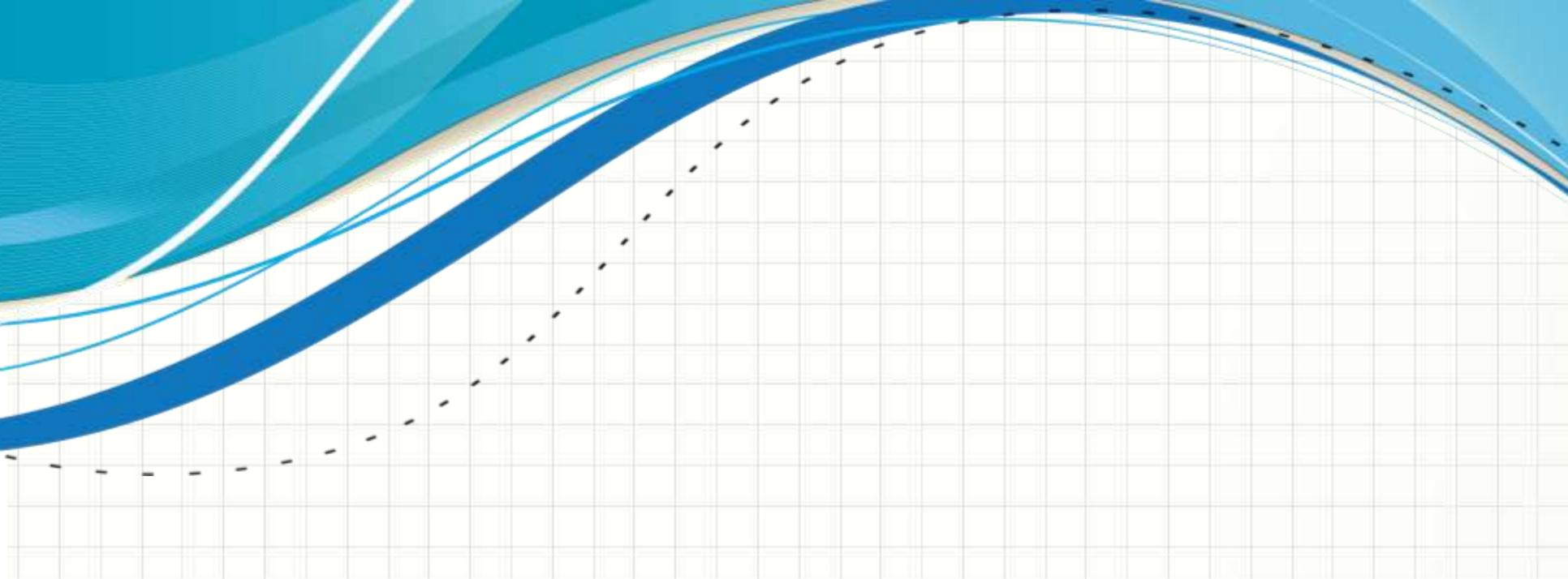
```
cout << "Digite um número: ";
```

```
cin >> A;
```

```
cout << "O número digitado foi: " << A;
```

```
getchar();
```

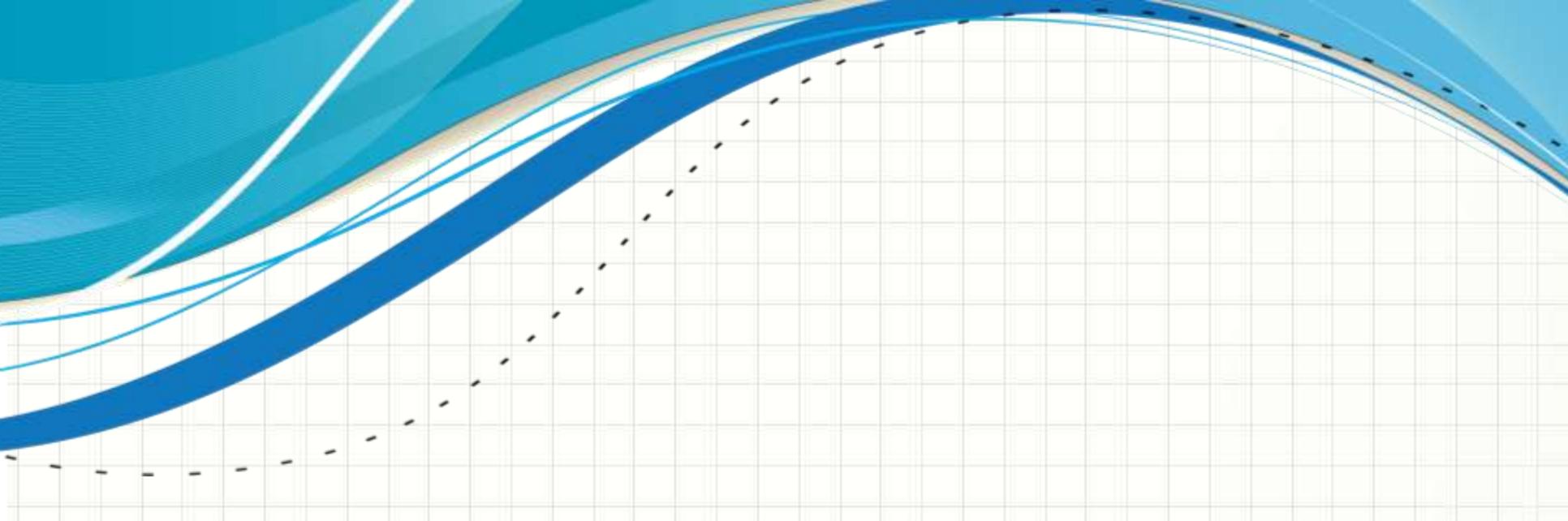
```
}
```



TUTORIAL

Tutorial

- Acompanhe o professor na criação de um programa “Olá Mundo!”
- Acompanhe o professor na criação de um programa “Calcula a média de 2 números”



CONCLUSÕES

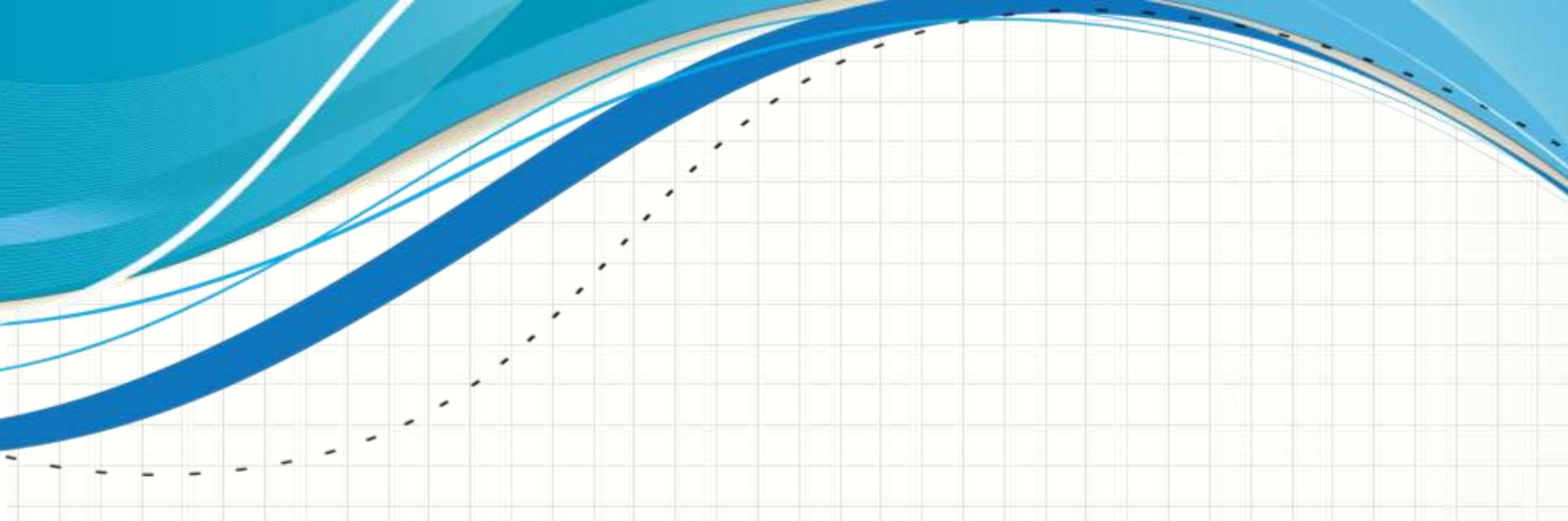
Resumo

- A linguagem C/C++ serve para que os algoritmos possam ser implementados no computador
- A linguagem C/C++ é mais rígida que o Portugol, impondo várias regras adicionais
- Por outro lado, a lógica de construção dos programas é exatamente a mesma!
- **TAREFA!**
 - Lista de Exercícios 1

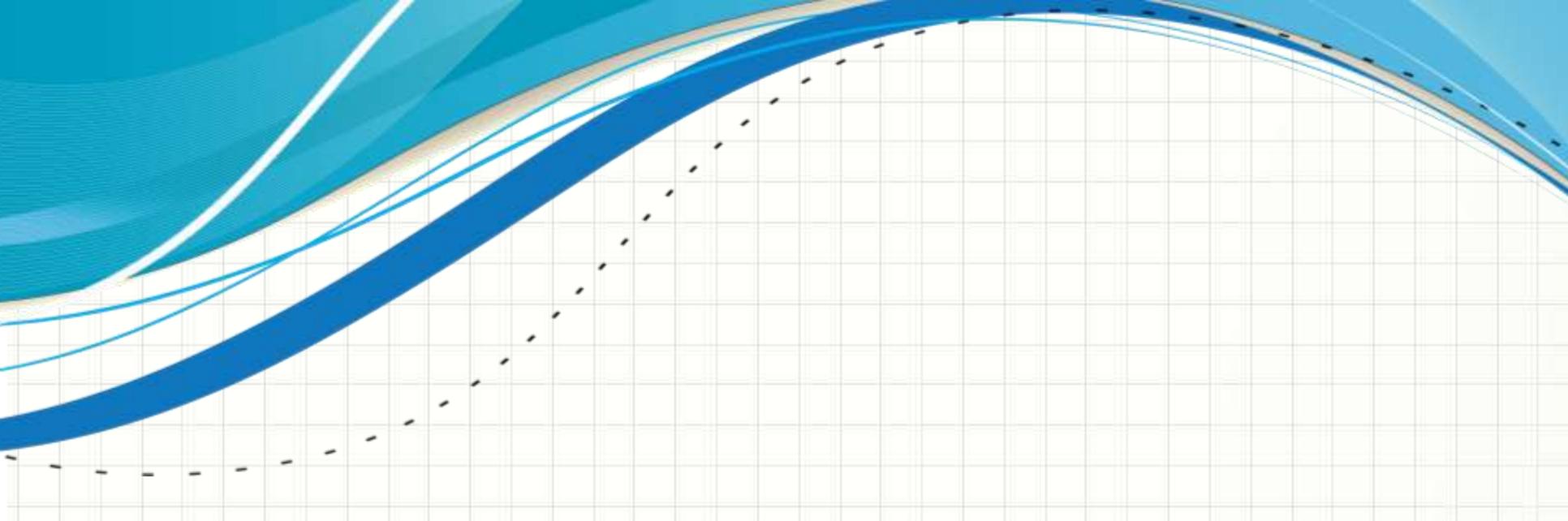
Próxima Aula



- O que mais o C/C++ pode fazer?
 - Ele é capaz de cálculos mais complexos?



PERGUNTAS?



**BOM DESCANSO
A TODOS!**