



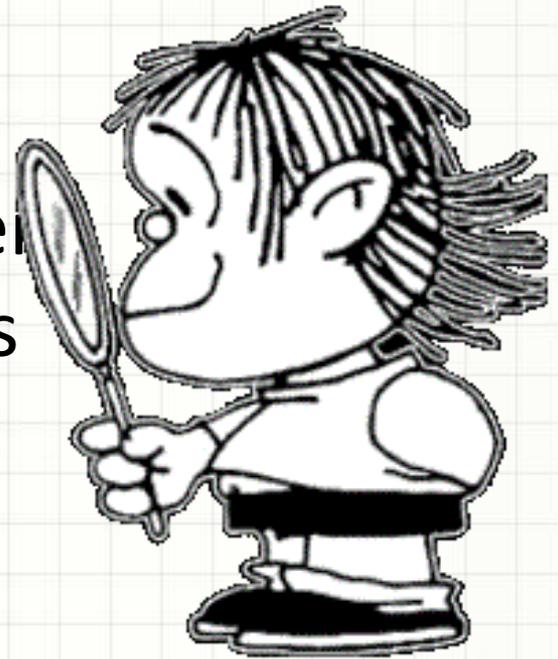
LÓGICA DE PROGRAMAÇÃO PARA ENGENHARIA INTRODUÇÃO À PROGRAMAÇÃO COM C/C++

Prof. Dr. Daniel Caetano

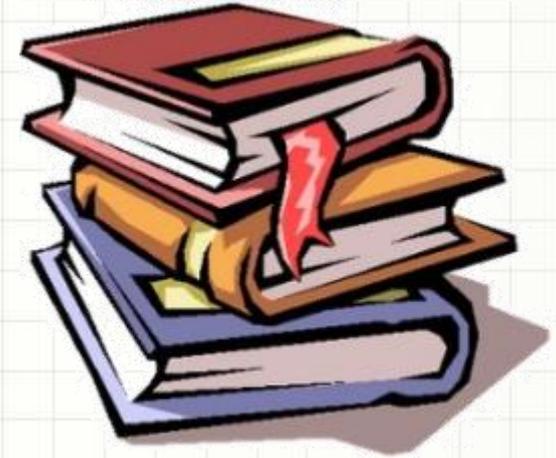
2013 - 1

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 Aula 5!**



Material de Estudo



Material

Acesso ao Material

Notas de Aula

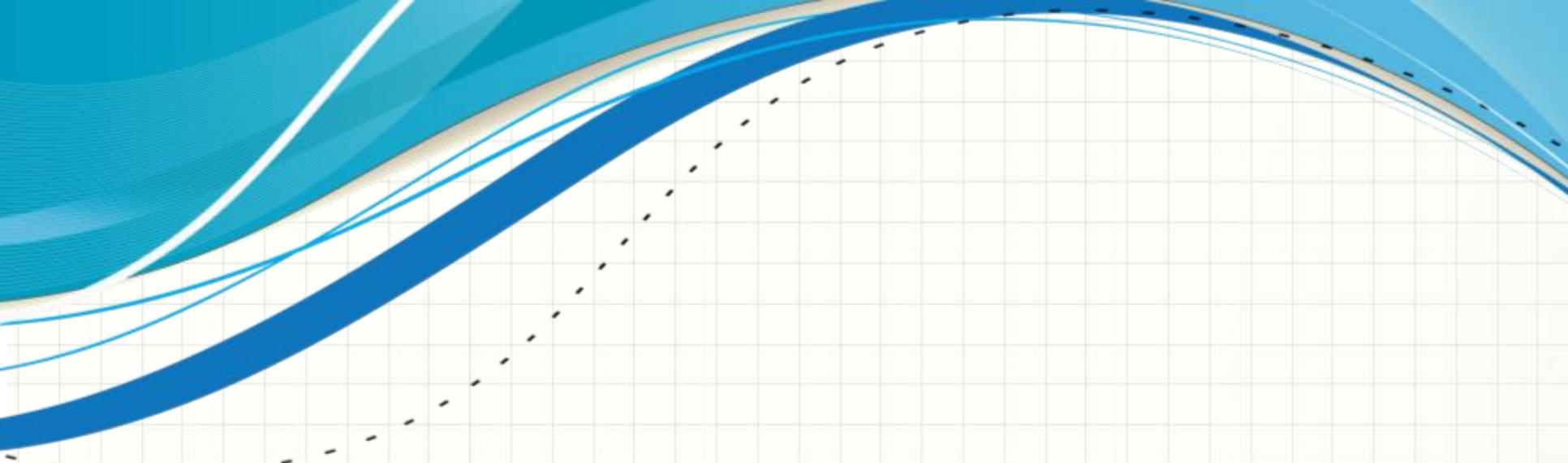
<http://www.caetano.eng.br/>
(Aula 5)

Apresentação

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

Material Didático

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



MIGRANDO PARA O C/C++

Por que C/C++?

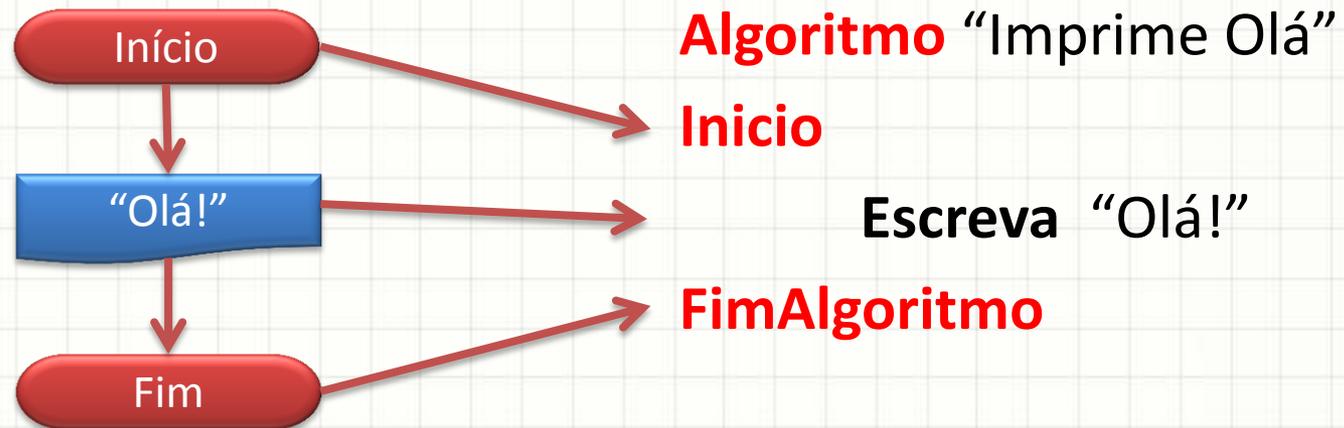
- Português Estruturado: Limitado
- Programas mais úteis: C/C++
 - Ver o resultado!
- Detalhamento:
 - Linguagem natural
 - Fluxograma
 - Português estruturado
 - Linguagem de Programação (C/C++)



**Detalhamento
Aumenta**

Algoritmo Mais Simples do Mundo

- Fluxograma
- Portugol



Algoritmo Mais Simples do Mundo

- Portugol

- Linguagem C

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

Algoritmo “Imprime Olá” → **main()**

Inicio → {

Escreva “Olá!” → **cout << “Olá!”;**

FimAlgoritmo → }

Algoritmo Mais Simples do Mundo

- Portugol

- Linguagem C

Algoritmo “Imprime Olá”

Início

Escreva “Olá”

FimAlgoritmo

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

main()

```
{
    cout << “Olá!”;
}
```

Área de Declarações

Algoritmo Mais Simples do Mundo

- Portugol

Algoritmo "Imprime Olá"

Inicio

Escreva "Olá!"

FimAlgoritmo

- Linguagem C

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

```
main()
```

```
{  
    cout << "Olá!";  
}
```

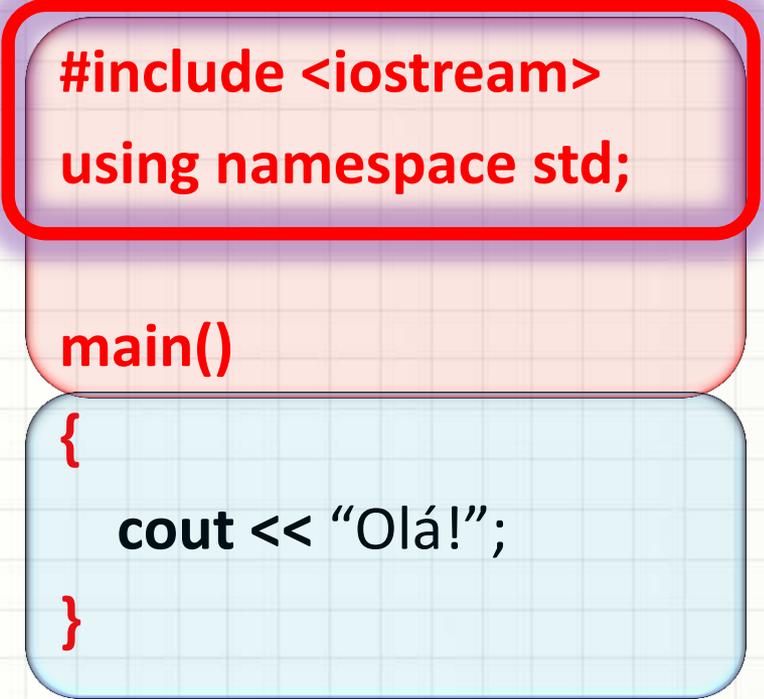
Área de Código

Algoritmo Mais Simples do Mundo

- Portugol



- Linguagem C



O que é?

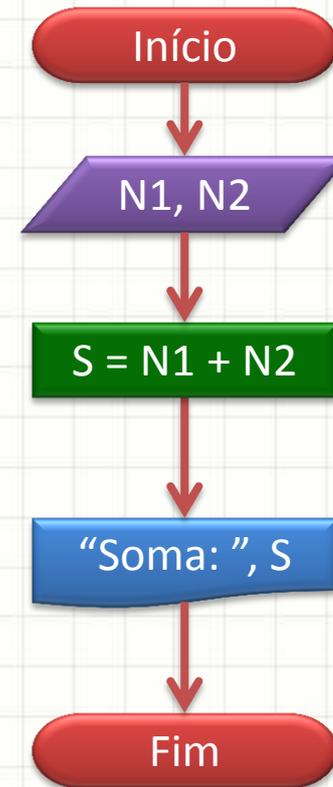
Area de Código

Somando Dois Números

- Linguagem Natural

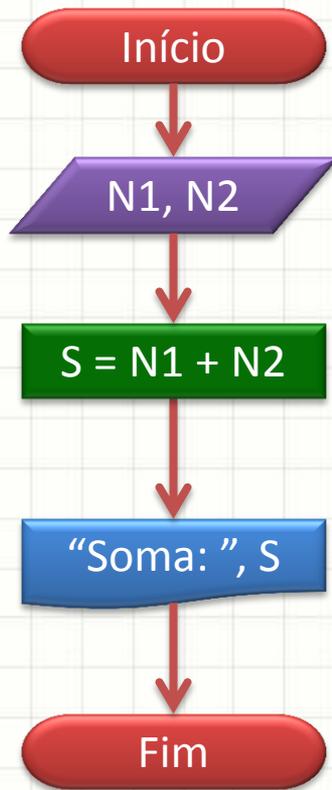
1. Leia dois números
2. Some os dois números
3. Imprima a soma

- Fluxograma



Somando Dois Números

- Fluxograma



- Portugol

Algoritmo "Soma 2 Números"

Inicio

Inteiro N1, N2, S

Escreva "Digite um No:"

Leia N1

Escreva "Digite outro No:"

Leia N2

$S \leftarrow N1 + N2$

Escreva "Soma:", S

FimAlgoritmo

Somando Dois Números

- Portugol

Algoritmo “Soma 2 Números”

Início

Inteiro N1, N2, S

Escreva “Digite um No:”

Leia N1

Escreva “Digite outro No:”

Leia N2

$S \leftarrow N1 + N2$

Escreva “Soma:”, S

FimAlgoritmo

- Linguagem C

```
#include <iostream>
```

```
using namespace std;
```

```
main()
```

```
{
```

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

```
    cout << “Digite um No:”;
```

```
    cin >> N1;
```

```
    cout << “Digite outro No:”;
```

```
    cin >> N2;
```

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

```
    cout << “Soma: ” << S;
```

```
}
```

Somando Dois Números

- Portugol

Algoritmo "Soma 2 Números"

Início

Inteiro N1, N2, S

Escreva "Digite um No:"

Leia N1

Escreva "Digite outro No:"

Leia N2

$S \leftarrow N1 + N2$

Escreva "Soma:", S

FimAlgoritmo

- Linguagem C

```
#include <iostream>
```

```
using namespace std;
```

```
main()
```

```
{
```

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

```
cout << "Digite um No:";
```

```
cin >> N1;
```

```
cout << "Digite outro No:";
```

```
cin >> N2;
```

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

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

```
}
```

Somando Dois Números

- Portugol

Algoritmo "Soma 2 Números"

Início

Inteiro N1, N2, S

Escreva "Digite um No:"

Leia N1

Escreva "Digite outro No:"

Leia N2

$S \leftarrow N1 + N2$

Escreva "Soma:", S

Fim

- Linguagem C

```
#include <iostream>
```

```
using namespace std;
```

```
main()
```

```
{
```

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

```
cout << "Digite um No:";
```

```
cin >> N1;
```

```
cout << "Digite outro No:";
```

```
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”

Inicio

Real DIAS, NS

Escreva “Quantos dias?”

Leia DIAS

$NS \leftarrow DIAS / 7$

Escreva “Semanas: ”, NS

FimAlgoritmo

Calculando o Número de Semanas

- Portugol

Algoritmo “Número de Semanas”

Início

Real DIAS, NS

Escreva “Quantos dias?”

Leia DIAS

NS ← DIAS / 7

Escreva “Semanas: ”, NS

FimAlgoritmo

- Linguagem C

```
#include <iostream>
```

```
using namespace std;
```

```
main()
```

```
{
```

```
float DIAS, NS;
```

```
cout << “Quantos dias?”;
```

```
cin >> DIAS;
```

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

```
cout << “Semanas:” << NS;
```

```
}
```

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

- Portugol

Algoritmo “Calcula IMC”

Início

Real P, A, IMC

Escreva “Digite peso (Kg):”

Leia P

Escreva “Digite altura (m):”

Leia A

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

Escreva “IMC: ”, IMC

FimAlgoritmo

Índice de Massa Corporal

- Portugol

Algoritmo “Calcula IMC”

Início

Real P, A, IMC

Escreva “Digite peso (Kg):”

Leia P

Escreva “Digite altura (m):”

Leia A

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

Escreva “IMC: ”, IMC

FimAlgoritmo

- Linguagem C

```
#include <iostream>
```

```
using namespace std;
```

```
main()
```

```
{
```

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

```
cout << “Digite peso (Kg): ”;
```

```
cin >> P;
```

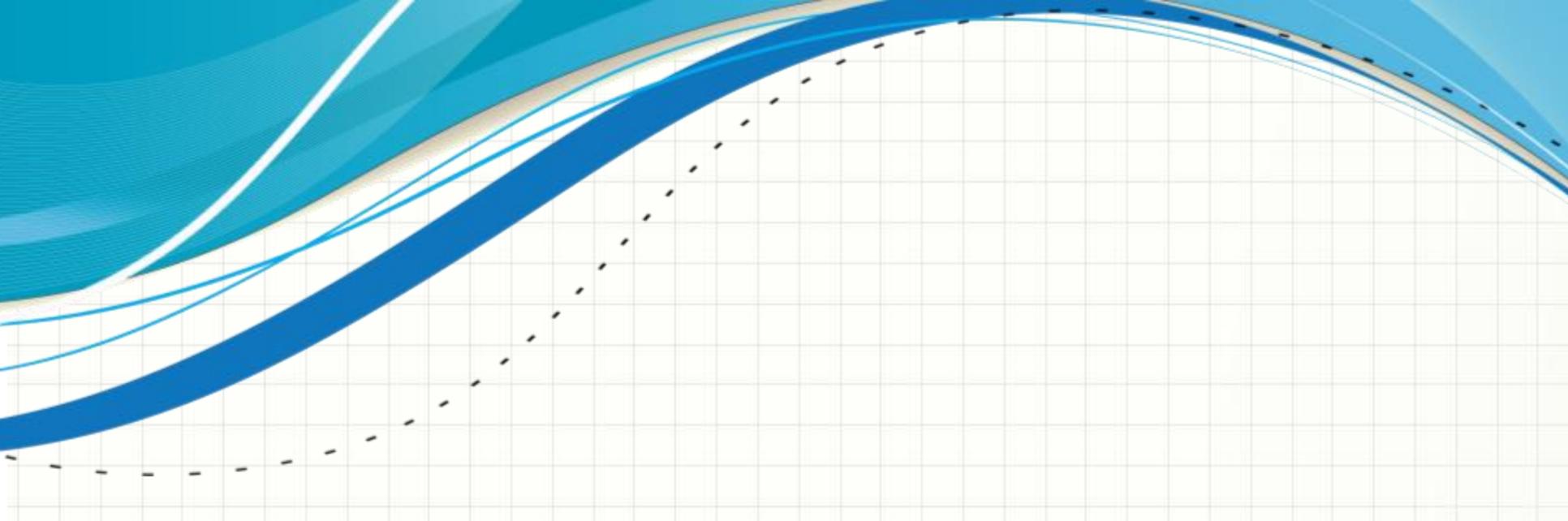
```
cout << “Digite altura (m):”;
```

```
cin >> A;
```

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

```
cout << “IMC: ” << IMC;
```

```
}
```



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 <iostream>  
using namespace std;
```

```
main()
```

```
{
```

```
    cout << "Olá!";
```

```
}
```

Estrutura do Programa

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

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

```
main()  
{
```

```
    cout << "Olá!";
```

```
}
```

Estrutura do Programa

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

;

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

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

Estrutura do Programa

- Vamos declarar uma variável A para INTEIROS

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

```
main()
```

```
{
```

```
    int A;
```

```
}
```

A declaração de variáveis tem sempre o seguinte formato:

TipoDaVariável NomeDaVariável ;

```
int main() {  
    using namespace std;
```

```
    main
```

```
{
```

```
    int A;
```

```
}
```

Em C/C++ não há
“:” entre o TIPO e o
NOME, há apenas
um espaço

Estrutura do Programa

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

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

```
main()  
{  
    int A, B;  
  
}
```

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;
```

```
main()  
{
```

```
    int A, B;
```

```
}
```

Estrutura do Programa

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

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

```
main()  
{  
    int A, B;  
    float C;  
}
```

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

```
main()
```

```
{
```

```
int A, B;
```

```
float C;
```

```
}
```

Estrutura do Programa

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

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

```
main()  
{  
    int A, B;  
    float C;  
  
}
```

Simplificadamente, os tipos de variáveis são:

int (inteiros)

float (reais)

bool (lógicos)

char/string (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)

```
main  
{  
    int A, B;  
    float C;  
}
```

Área de Código

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

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

```
main()  
{
```

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

```
}
```

Área

Uma frase qualquer

- Tamanho
- Escrita

a

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

```
}
```

Área de Código

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

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

```
main()  
{
```

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

```
}
```

Área

Uma frase qualquer1934

- Es
- E

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

```
cout << 1934;
```

```
}
```

Área de Código

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

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

```
main()  
{
```

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

```
    cout << endl;
```

```
    cout << 1934;
```

```
}
```

Área

Uma frase qualquer
1934

- Es
- E

inha



```
cout << "Uma frase qualquer";  
cout << endl;  
cout << 1934;
```

```
}
```

Área de Código

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

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

```
main()  
{
```

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

```
}
```

Área

$$2 + 3 = 5$$

- Es
- E

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

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

```
}
```

Área

2 + 3 = 5

- Es
- E e

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

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

}

Área

2 + 3 = 5

- Es
- E e

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

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

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

```
}
```

Área de Código

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

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

```
main()  
{
```

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

```
}
```

Área de Código

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

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

```
main()  
{
```

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

```
}
```

Áre

$$2 + 3 = 5$$

- N/A
se

só,

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

```
}
```

Nas expressões, podemos usar as operações matemáticas usuais:

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

```
main()
{
```



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

```
}
```

Área de Código

- Para fazer coisas mais elaboradas, usamos as variáveis

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

```
main()
```

```
{
```

```
    int SOMA;
```

```
    SOMA = 2 + 3;
```

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

```
}
```

Área de Código

- Para fazer coisas mais interessantes com variáveis

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

```
int main()
```

```
{
```

```
    int SOMA;
```

```
    SOMA = 2 + 3;
```

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

```
}
```

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

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

```
using namespace std;
```

```
main()
```

```
{
```

```
    int SOMA;
```

```
    SOMA = 2 + 3;
```

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

```
}
```

...e um valor ou expressão
do lado direito

```
using namespace std;
```

```
main()
```

```
{
```

```
int SOMA;
```

```
SOMA = 2 + 3;
```

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

```
}
```



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

```
using namespace std,
```

```
main()
```

```
{
```

```
int SOMA,
```

```
SOMA = 2 + 3;
```

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

```
}
```



Isso significa que o computador resolve a linha destacada assim:

SOMA = 2+3;

SOMA = 5;

`using namespace std;`

`main()`

`{`

`int SOMA;`

`SOMA = 2 + 3;`

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

`}`



Área de Código

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

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

```
main()
```

```
{
```

```
    int A;
```

```
    cin >> A;
```

```
}
```

Área de Código

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

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

```
main()  
{
```

```
    int A;
```

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

```
    cin >> A;
```

```
}
```

Área

- Variáveis

Digite um número: _

e o

```
int A;
```

```
cout << "Digite um número: ";
```

```
cin >> A;
```

```
}
```

Área de Código

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

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

```
int main()
```

```
{
```

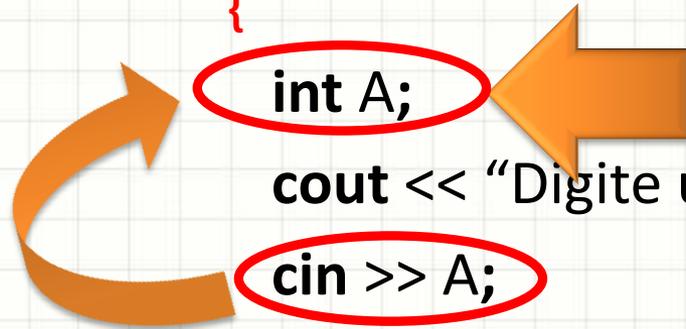
```
    int A;
```

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

```
    cin >> A;
```

```
}
```

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



Área de Código

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

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

```
main()  
{
```

```
    int A;
```

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

```
    cin >> A;
```

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

```
}
```

Área

- Ag
im

Digite um número: 1357
O número digitado foi: 1357

mos

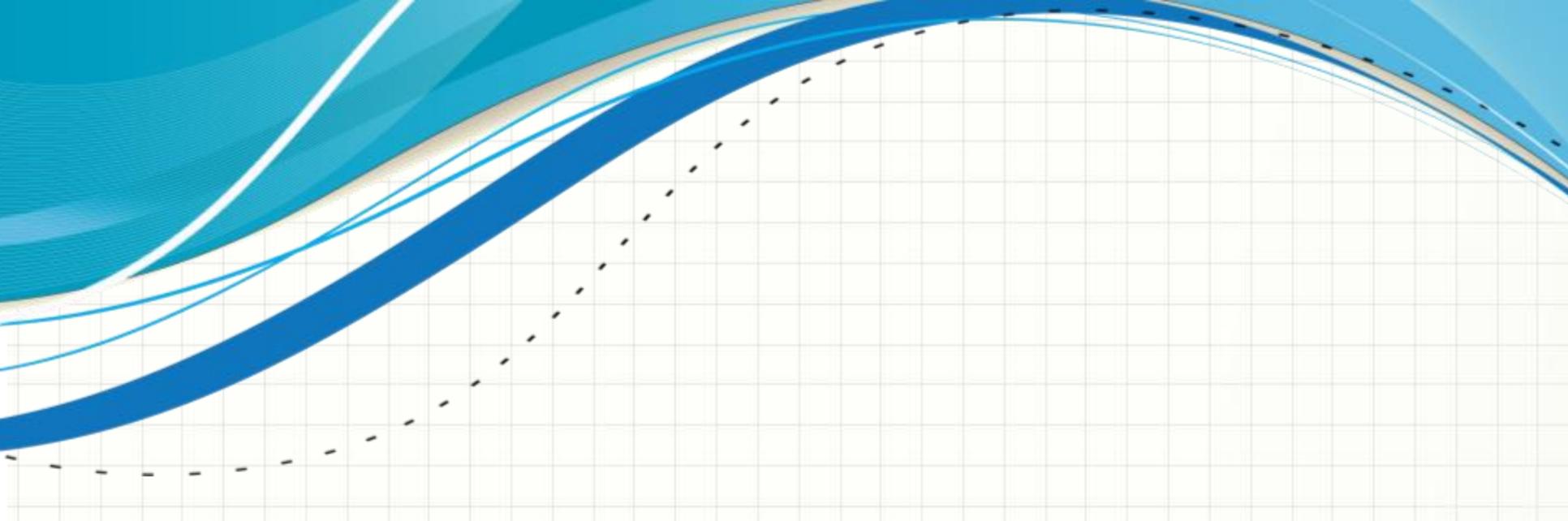
```
int A;
```

```
cout << "Digite um número: ";
```

```
cin >> A;
```

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

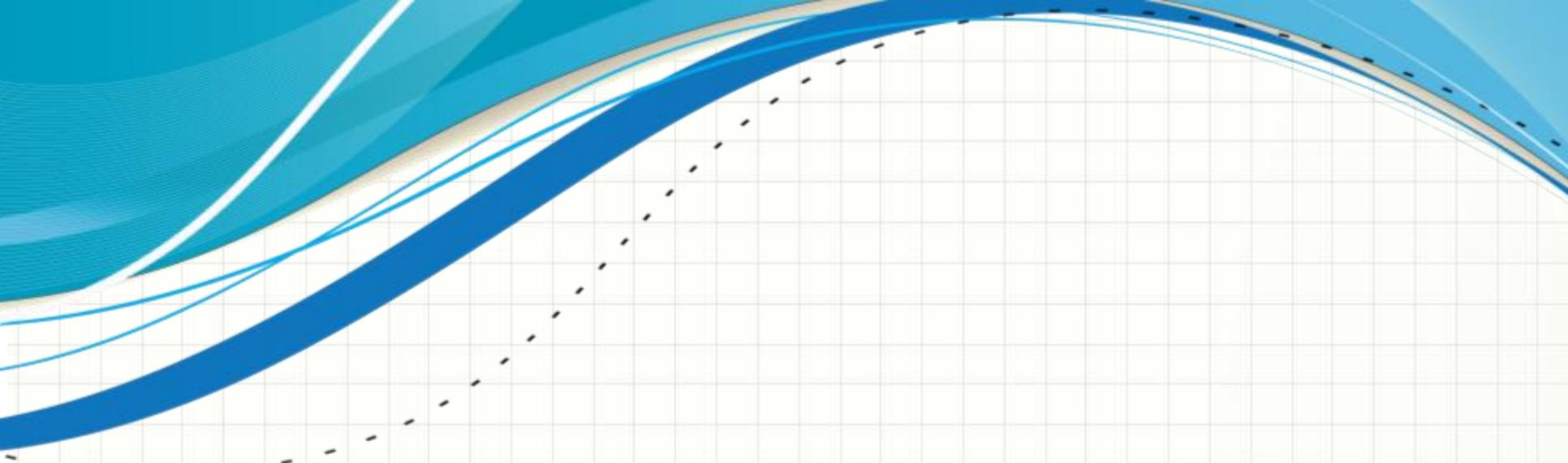
```
}
```



CONVERTENDO PORTUGOL PARA C/C++

Tabela Prática de Conversão

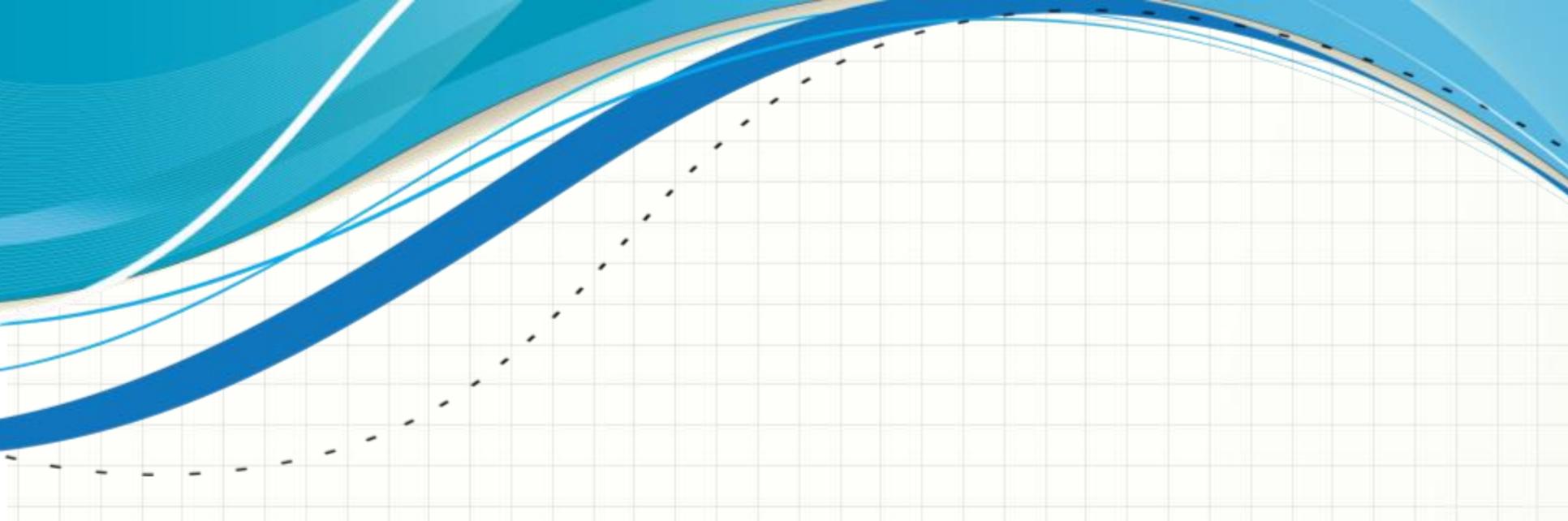
Objetivo	Portugol	C/C++
Declarar variável inteira	inteiro x	int x;
Declarar variável “com vírgula”	real x	float x;
Imprimir texto	escreva “Texto”	cout << “Texto”;
Imprimir texto pulando linha	escreva “Texto”, fiml	cout << “Texto” << endl;
Imprimir valor de variável	escreva x	cout << x;
Guardar valor em variável	x <- 5	x = 5;
Guardar resultado em variável	x <- 2*y	x = 2*y;
Ler um valor do teclado	leia x	cin >> x;
Marcar início do código	inicio	{
Marcar fim do código	fimalgoritmo	}



TUTORIAL

Tutorial

- Acompanhe o professor na criação de um programa “Bem vindo ao C!”
- Acompanhe o professor na criação de um programa “Calcula a média de 2 números”



CONCLUSÕES

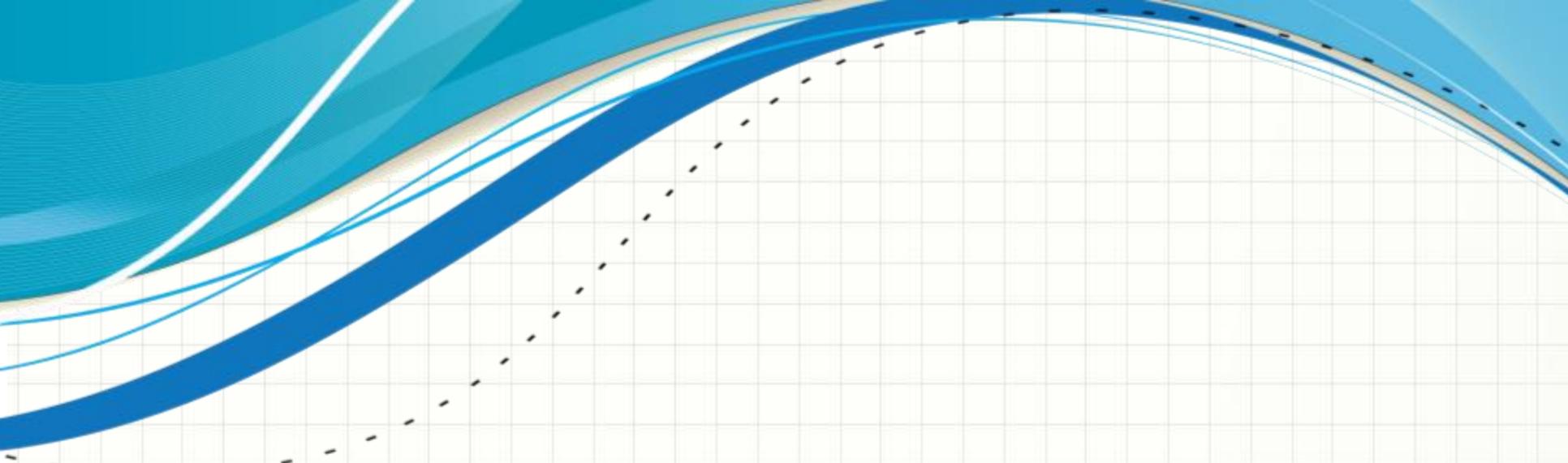
Resumo

- 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 Aula 5!

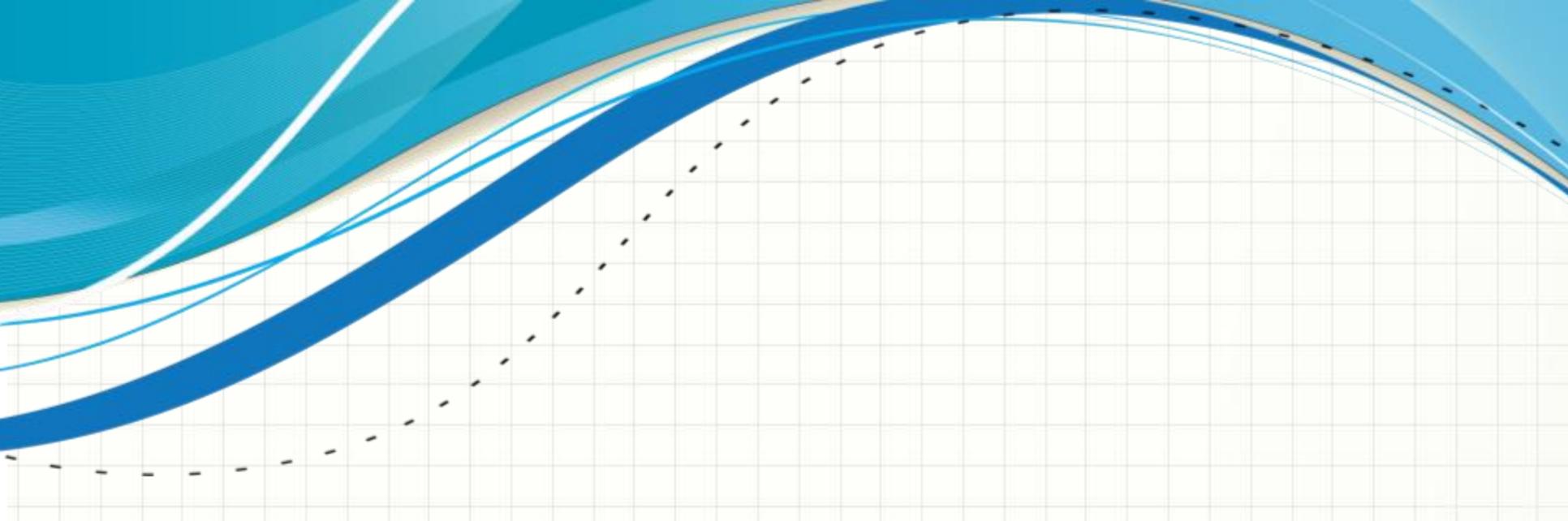
Próxima Aula



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



PERGUNTAS?



**BOM DESCANSO
A TODOS!**