



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

Prof. Dr. Daniel Caetano

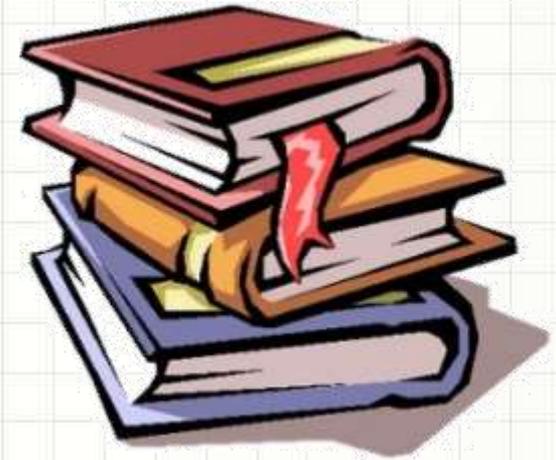
2018 - 1

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++
- **Atividades Aula 5 – SAVA!**



Material de Estudo



Material

Acesso ao Material

Notas de Aula e
Apresentação

<http://www.caetano.eng.br/>
(Lógica de Programação para Eng. – Aula 5)

Material Didático

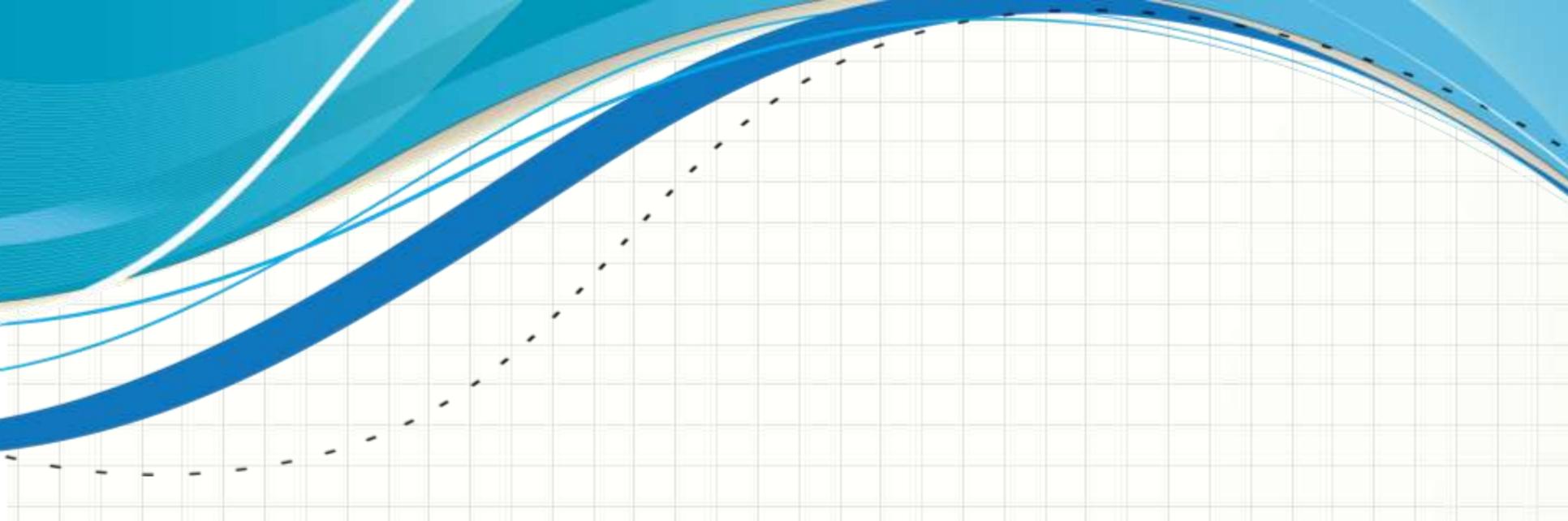
Lógica de Programação, págs 42 a 68, 104 a 117.

Aula Online

Aula 3

Biblioteca Virtual

“Lógica de Programação – Fundamentos da
Programação de Computadores”, págs 7 a 49.



MIGRANDO PARA O C/C++

Por que C/C++?

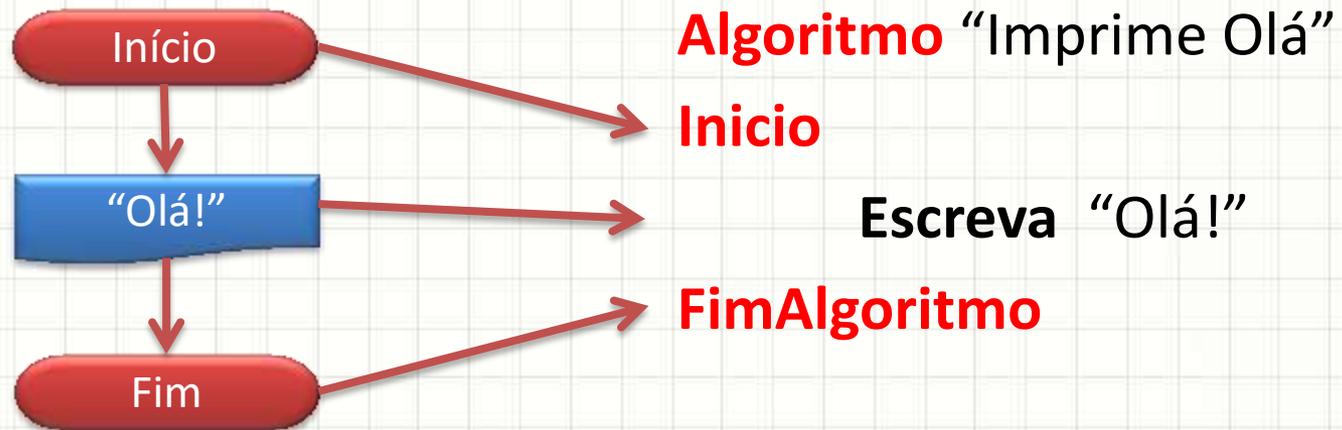
- Português Estruturado: Limitado
- Programas mais úteis: C/C++
 - Ver o resultado!
- Por outro lado...
 - Linguagem natural
 - Fluxograma
 - Português estruturado
 - Linguagem de Programação (C/C++)



**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()** // Imprime Olá

Inicio → {

Escreva “Olá!” → **cout** << “Olá!”;

FimAlgoritmo → }

Algoritmo Mais Simples do Mundo

- Portugol

O que é?

Algoritmo “Imprime Olá”

Inicio

Escreva “Olá!”

FimAlgoritmo

- Linguagem C

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

```
main() // Imprime Olá
```

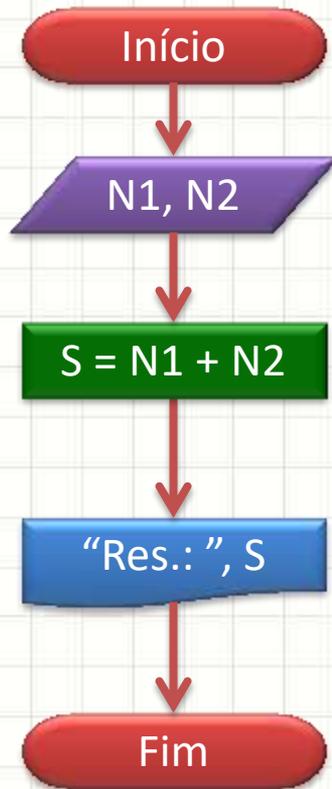
```
{
```

```
    cout << “Olá!”;
```

```
}
```

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 ← 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() // Soma 2 Números
```

```
{
```

```
    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() // Soma 2 Números
```

```
{
```

```
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() // Soma 2 Números
```

```
{
```

```
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

Convertendo Dias para Semanas

- Linguagem Natural
 1. Leia o número de dias
 2. Divida o número de dias por 7
 3. Imprima o número de semanas

- Fluxograma



Convertendo Dias para Semanas

- Fluxograma



- Portugol

Algoritmo “Número de Semanas”

Inicio

Real DIAS, NS

Escreva “Quantos dias?”

Leia DIAS

$NS \leftarrow DIAS / 7$

Escreva “Semanas: ”, NS

FimAlgoritmo

Convertendo Dias para Semanas

- Portugol

Algoritmo “Número de Semanas”

Inicio

Real DIAS, NS

Escreva “Quantos dias?”

Leia DIAS

NS ← DIAS / 7

Escreva “Semanas: ”, NS

FimAlgoritmo

- Linguagem C

```
#include <iostream>
```

```
using namespace std;
```

```
main() // Número de Semanas
```

```
{
```

```
float DIAS, NS;
```

```
cout << “Quantos dias?”;
```

```
cin >> DIAS;
```

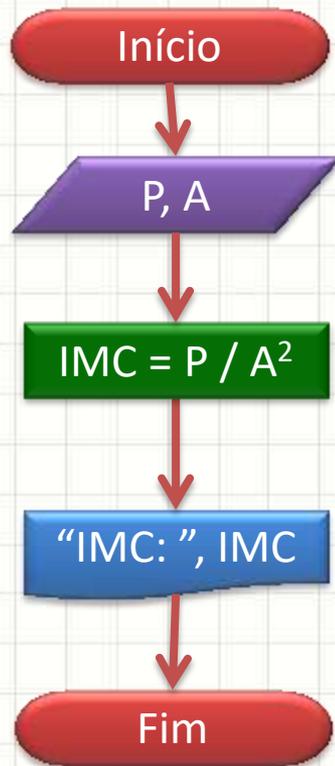
```
NS = DIAS / 7;
```

```
cout << “Semanas:” << NS;
```

```
}
```

Índice de Massa Corporal

- Fluxograma



- Portugol

Algoritmo "Calcula IMC"

Inicio

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() // Calcula IMC
```

```
{
```

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

```
    cout << “Digite peso (Kg): ”;
```

```
    cin >> P;
```

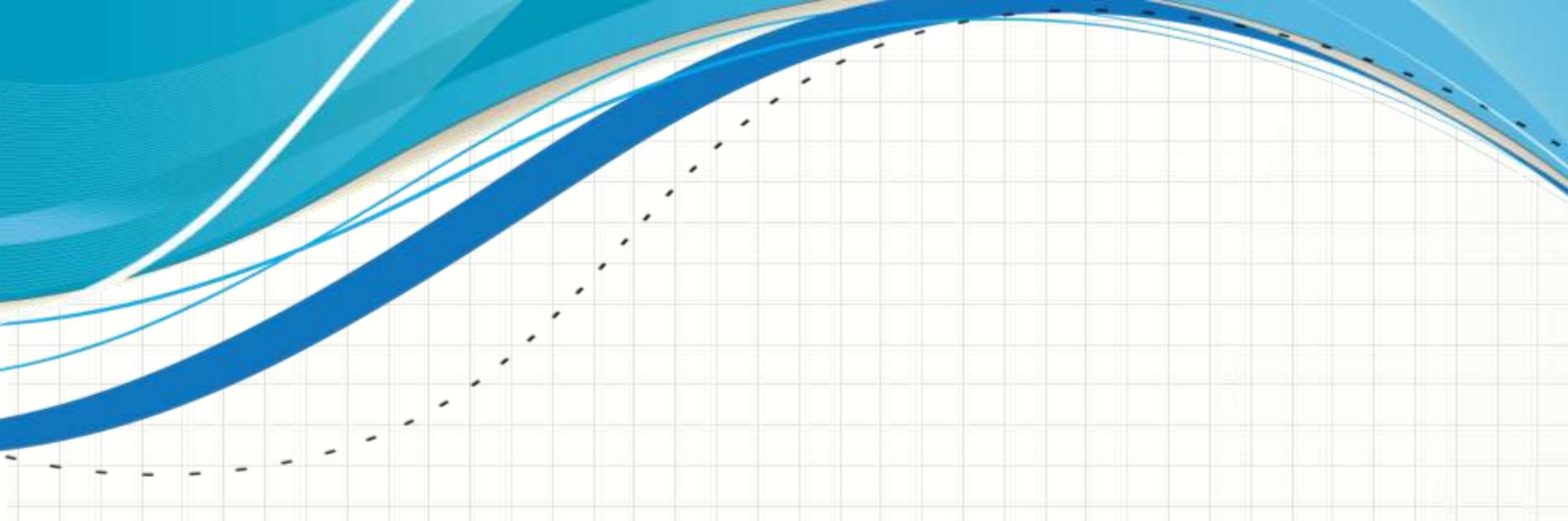
```
    cout << “Digite altura (m):”;
```

```
    cin >> A;
```

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

```
    cout << “IMC: ” << IMC;
```

```
}
```

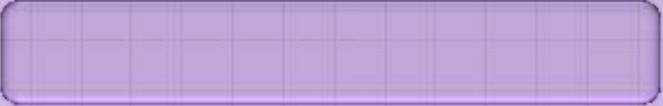


A LINGUAGEM C/C++

Estrutura do Programa

- Estrutura básica de um programa em C/C++:
- Há diferenciação: maiúsculas x minúsculas!
 - **main** ≠ **Main**
- A área de variáveis
 - Primeira coisa!
- Nome do programa
 - Sempre **main**
- Comentários
 - Tudo depois de **//**

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

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

Estrutura do Programa

- Estrutura básica de um programa em C/C++:
- Área do algoritmo...
 - Demarcada por { ... }
- Linhas de código
 - Terminadas com

;

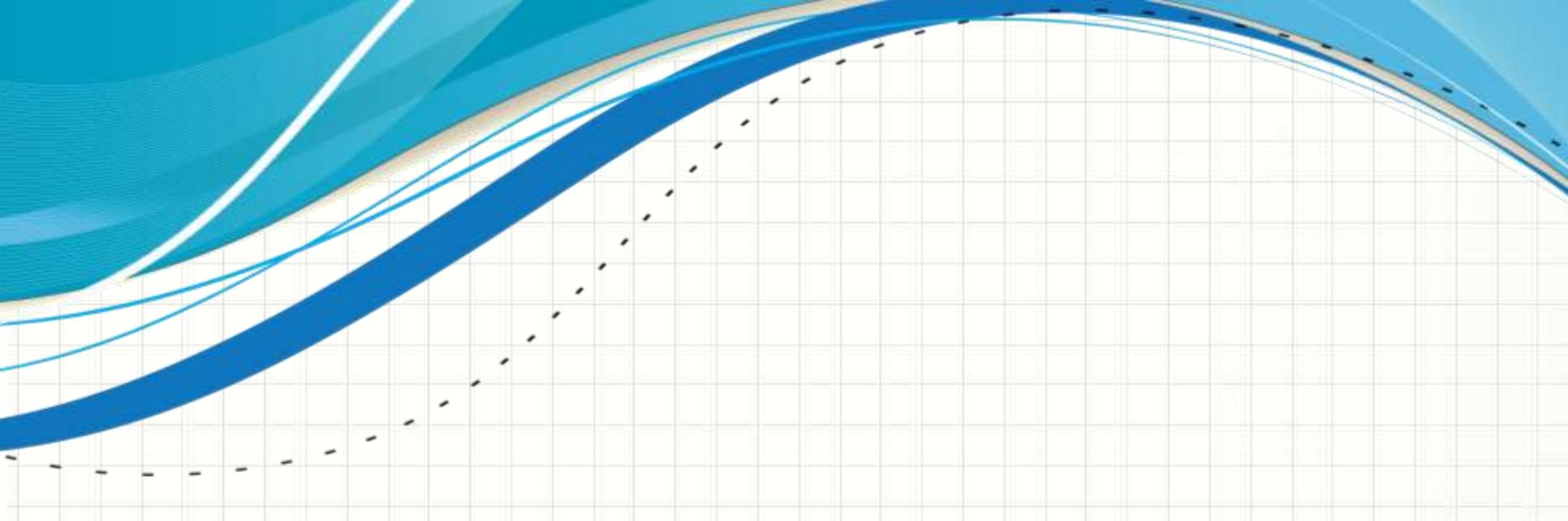
```
#include <iostream>  
using namespace std;
```

```
main() // Nome Legível
```

```
{
```

```
cout << "Olá!";
```

```
}
```



**MOSTRANDO
INFORMAÇÕES**

Mostrando Informações

- Como apresentar texto na tela?
- Escrever “Uma frase qualquer”

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

```
main() // Imprime uma frase  
{
```

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

```
}
```

Mo

- Co
- Es

Uma frase qualquer

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

```
}
```

Mostrando Informações

- Escrevendo várias coisas...

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

```
main() // Escrevendo várias informações  
{
```

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

```
}
```

Mo

Uma frase qualquer
1934

- Es

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

```
}
```

Mostrando Informações

- Diferença de *strings* e *expressões*

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

main() // Strings e expressões
{

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

}
```

Mc

2 + 3 = 5

- Di

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

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

```
}
```

Mostrando Informações

- Economizando linhas de código...

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

```
main() // Impressão múltipla  
{
```

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

```
}
```

2 + 3 = 5

Mo

- Ec

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

```
cout << "O peso é: " << (50*1.5) << " kg";
```

Mostrando Informações

Operadores:

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

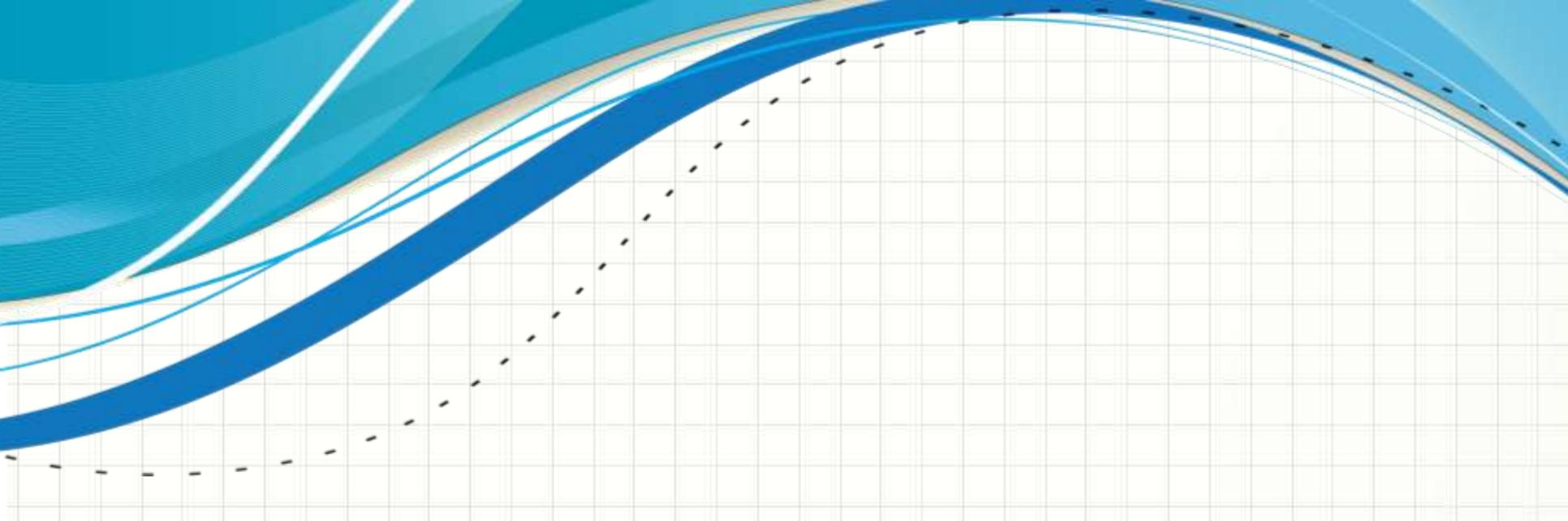
```
main() // Impressão múltipla
```

```
{
```

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

```
}
```





USANDO VARIÁVEIS

Usando variáveis

- O salário por hora é R\$ 20.50. O funcionário trabalhou 12 horas. Quanto ele deve receber?

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

```
main() // Calculando Salário
```

```
{
```

```
float SALARIO;
```

Variável

```
SALARIO = 20.50 * 12.0;
```

Atribuição

```
cout << SALARIO;
```

```
}
```

Usando Variáveis

- Declarando a variável **A** para INTEIROS

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

```
main() // Declarando variáveis  
{  
    int A;  
}
```

Usando Variáveis

A declaração de variáveis tem sempre o seguinte formato:

TipoDaVariável NomeDaVariável ;

```
using namespace std;
```

```
main() // Declarando variáveis
```

```
{
```

```
    int A;
```

```
}
```

Usando Variáveis

- Declarando a variável **A** para INTEIROS
- Declarando variável **B** também para INTEIROS

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

```
main() // Declarando variáveis  
{  
    int A, B;  
}
```

Usando Variáveis

- Além de A e B, declarando **C** para REAIS...

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

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

Usando Variáveis

- Além de A e B, declarando C para REAIS...

```
#include <iostream>
```

```
using namespace std;
```

```
int main() // Dec
```

```
{
```

```
    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. SEM espaços
2. SEM caracteres especiais (acentos etc.)
3. Proibido o uso de palavras da linguagem
(float ou main, por exemplo)
4. Não iniciar com números

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

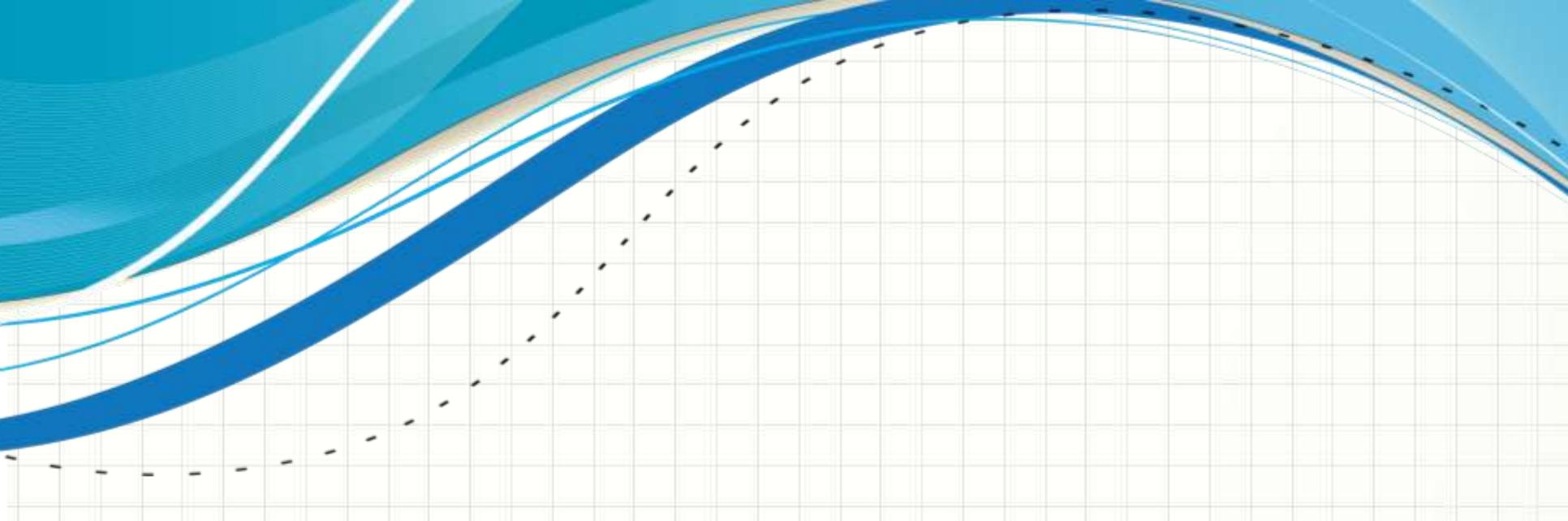
Atribuição de Valores

- O operador atribuição: =
- Guarda **resultado de um cálculo** na variável

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

```
main() // Atribuindo valores  
{  
    int A, B, C;  
    A = 5;  
    B = 5+10;  
    C = B - A;  
    cout << "Vals.:" << A << ", " << B ", " << C;  
}
```

Sempre é usado
com uma variável
à esquerda e um
valor ou expressão
à direita!



ENTRADA DE DADOS

Entrada de Dados

- Variáveis guardam valores
- Atribuições armazenam valores de cálculos
- E para **receber informações do usuário?**

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

```
main() // Entrada de Dados  
{  
    int A;  
  
    cin >> A;  
}
```

Congela o
programa
esperando o
usuário digitar algo

Enf

- V
- A
- E

los

O que o usuário deve digitar?

```
int A;
```

```
cin >> A;
```

```
}
```

Entrada de Dados

- É útil avisar ao usuário o que ele deve digitar

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

```
main() // Entrada de Dados
```

```
{
```

```
    int A;
```

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

```
    cin >> A;
```

```
}
```

Entrada de Dados

Digite um número: _

- É

gitar

```
cout << "Digite um número: ";
```

```
cin >> A;
```

```
}
```

Entrada de Dados

Digite um número: _

- É

O computador irá esperar até que o usuário digite algo e aperte a tecla enter

```
cout << "Digite um número: ";
```

```
cin >> A;
```

```
}
```

Entrada de Dados

- Tipo de informação depende da variável:

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

```
main() // Entrada de Dados
```

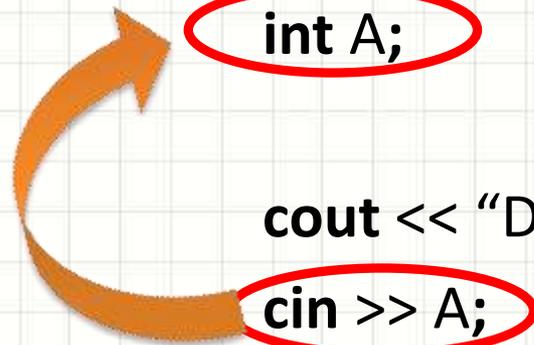
```
{
```

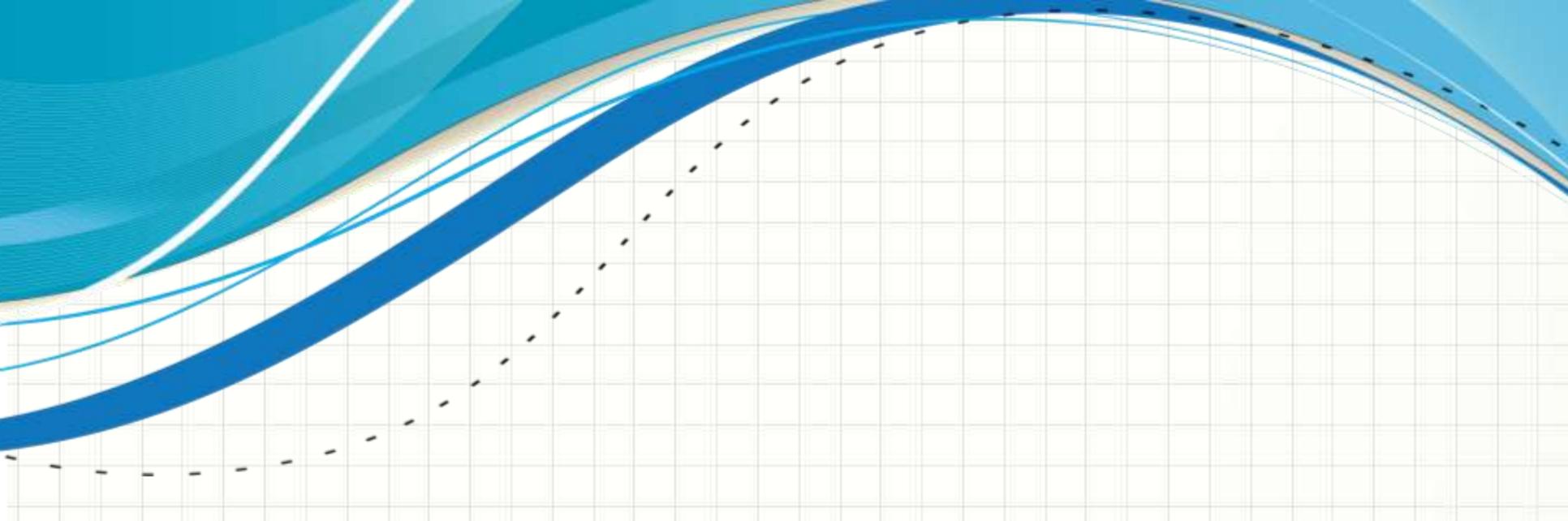
```
int A;
```

```
cout << "Digite um número:";
```

```
cin >> A;
```

```
}
```

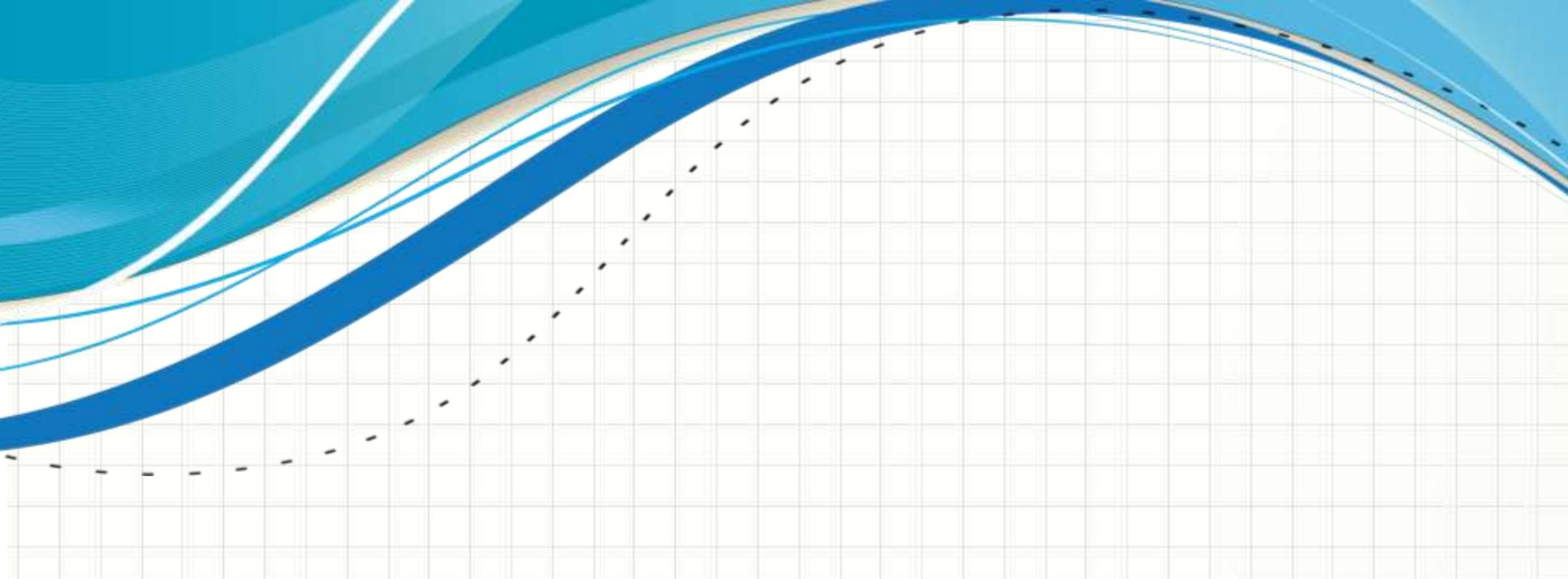




CONVERTENDO PORTUGOL PARA C/C++

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;
Declarar nome do algoritmo	Algoritmo “Nome”	main() // Nome
Marcar início do código	inicio	{
Marcar fim do código	fimalgoritmo	}



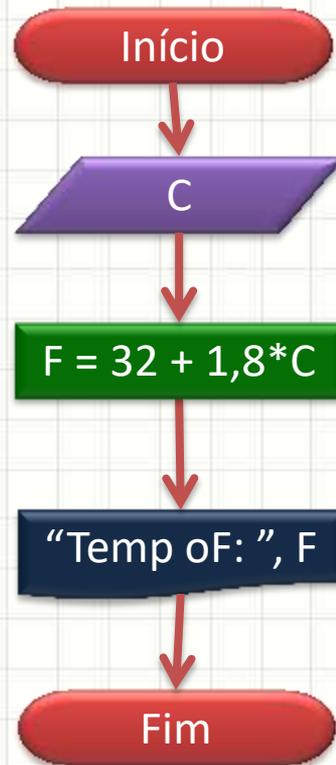
TUTORIAL

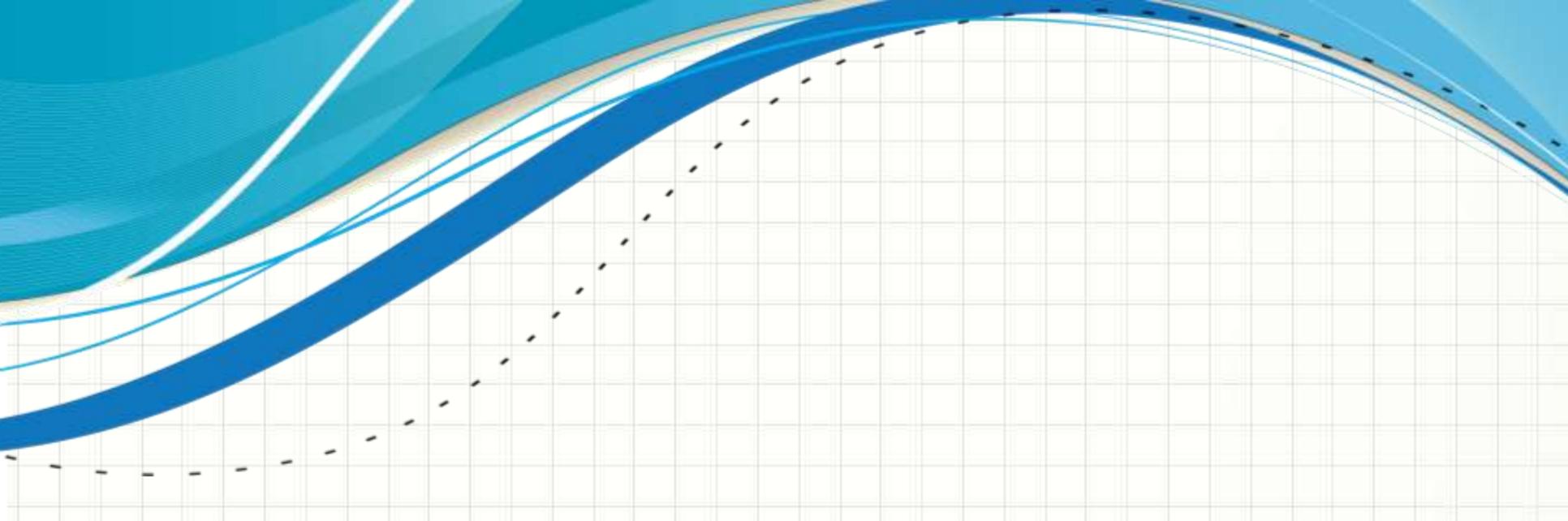
Tutorial

- 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”

Tutorial

- Como programar esse fluxograma abaixo?



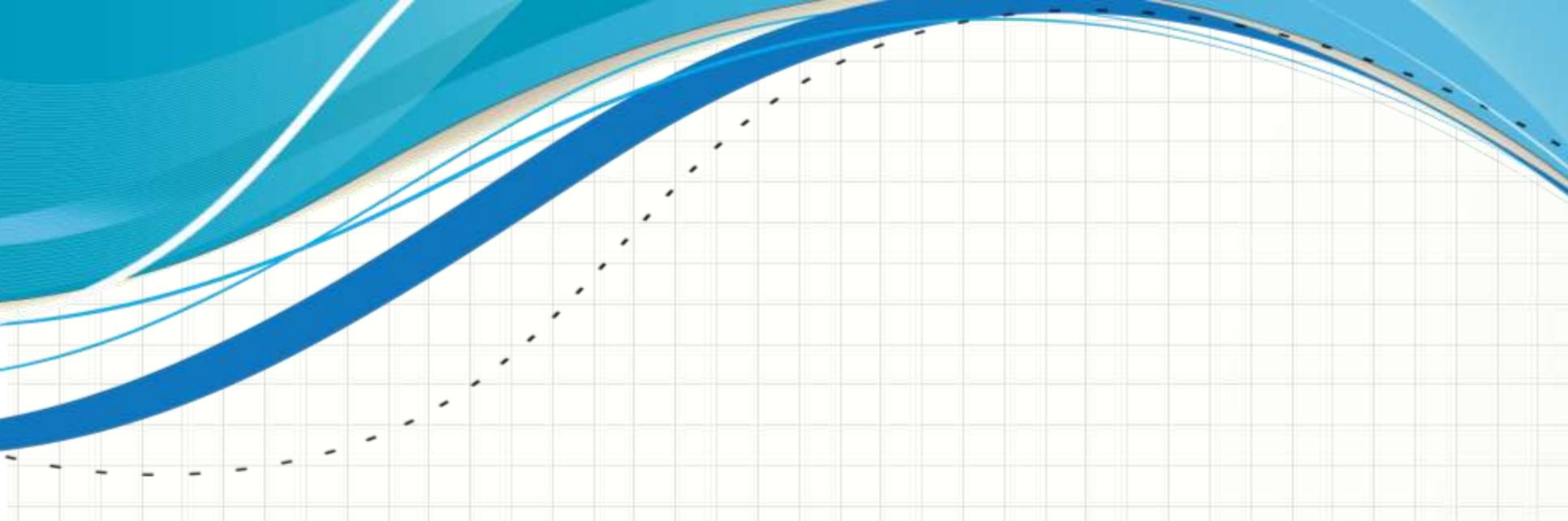


CONCLUSÕES

Resumo

- C/C++: implementar algoritmos “reais”
- C/C++: mais “rígido” que o Portugol
- Lógica de construção dos programas: idêntica!
- **TAREFA: Lista Aula 5!**

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



PERGUNTAS?