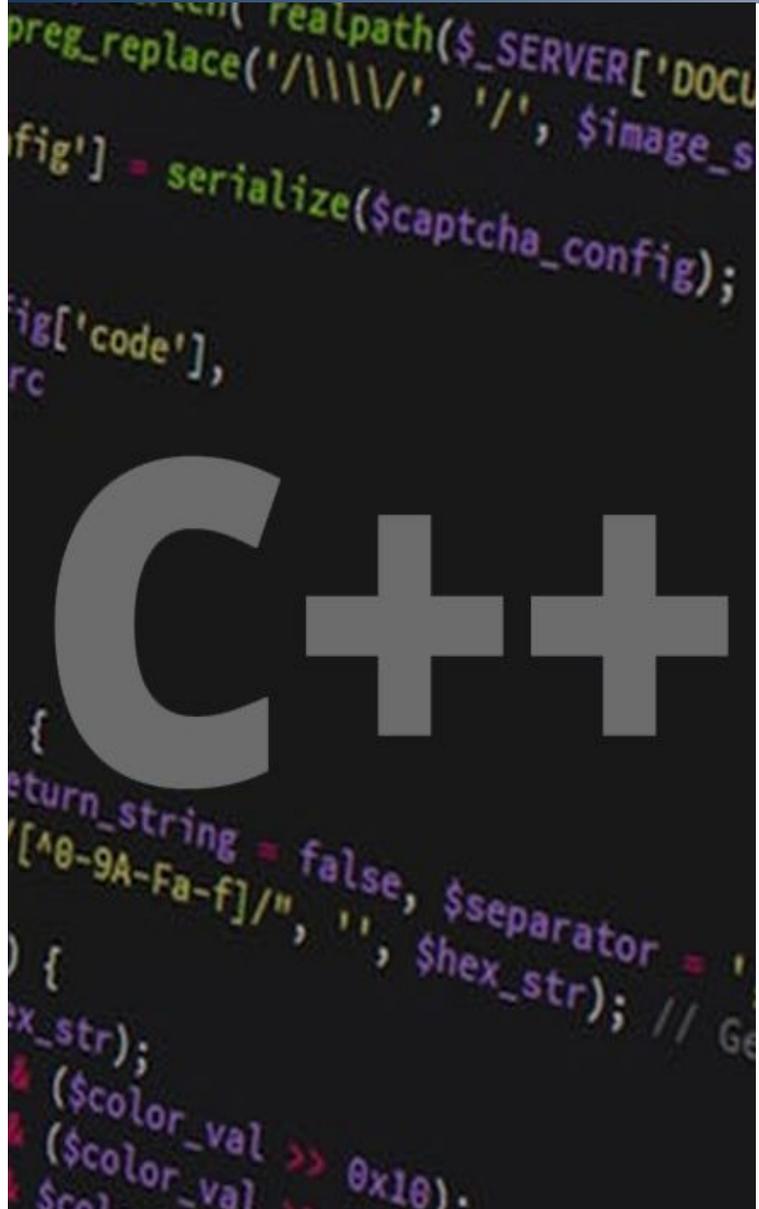




UFAC



# Linguagem de Programação II

Prof. André Nasserla  
andre.nasserla@ufac.br

# Estrutura Básica do Código C++

- A unidade fundamental de programas C++ são as funções.
- Um programa C++ consiste em uma ou várias funções.
- Os elementos básicos de toda função C++ são os seguintes:

```
tipo nome ()  
  
{  
  
    instrução_1 ;  
    instrução_2 ;  
    .  
    .  
    .  
    instrução_n ;  
  
}
```

# Estrutura Básica do Código C++

- Você pode digitar este programa e gravá-lo em disco com o nome “hello.cpp”.
- O sufixo .CPP indica programa-fonte em C plus plus.
- Vamos começar com um programa muito simples:

```
#include <iostream.h>

void main()
{
    cout << "Primeiro Programa" ;
}
```

# Estrutura Básica do Código C++

- **NOME DAS FUNÇÕES**

- Este programa compõe-se de uma única função chamada main.
- O nome de uma função pode ser qualquer um, com exceção de main, reservado para a função que inicia a execução do programa.
- Em todo programa C++, deve existir uma única função chamada main.
- A função main marca o ponto de partida do programa.
- Se um programa for constituído de uma única função, esta será main.
- O programa termina quando for encerrada a execução da função main.

# Estrutura Básica do Código C++

- **O TIPO void**
- A função main particular de nosso programa é do tipo void.
- O tipo void indica que a função não tem valor de retorno, ou seja, não retorna nada.
- Os tipos de funções serão abordados em detalhes mais adiante.
- **CHAVES**
- Toda função C++ deve começar com uma chave de abertura de bloco e deve terminar com uma chave de fechamento de bloco }.
- As chaves delimitam o corpo da função.

# Estrutura Básica do Código C++

- **ESPAÇOS EM BRANCO**
- Você pode inserir espaços em branco e tabulações e pular linhas à vontade em seus programas.
- O compilador ignora esses caracteres.
- Você pode escrever várias instruções em uma única linha, separadas por qualquer número de espaços ou tabulações, ou pode escrever uma instrução em várias linhas.
- Não existe um estilo obrigatório para a escrita de programas C++.
- O nosso primeiro programa poderia ser escrito da seguinte forma:

```
#include <iostream.h>

void main() {
    cout <<
        "Primeiro Programa"
        ;
}
```

# Estrutura Básica do Código C++

- A sintaxe anterior não é recomendada, pois esse estilo causa sérios problemas de legibilidade.
- Algumas exceções no uso de espaços, tabulações e linhas em branco devem ser consideradas:
  - cadeias de caracteres constantes como "Primeiro Programa" não podem ser separadas em diversas linhas;
  - nomes de funções, operadores e comandos da linguagem não podem ser separados.
  - Diretivas do pré-processador como #include não podem ser escritas em diversas linhas.

# Estrutura Básica do Código C++

- **ARQUIVOS DE INCLUSÃO**

- Os arquivos de inclusão (também chamados arquivos de cabeçalho) são textos escritos em caracteres ASCII normais.
- Em geral, eles contêm definições e declarações necessárias para que o compilador reconheça vários identificadores da linguagem C++.
- Você pode verificar o conteúdo desses arquivos utilizando o comando type do WINDOWS, ou cat do LINUX.
- Geralmente, os arquivos de inclusão têm um nome terminado com o sufixo ".H" (de header ou cabeçalho) e estão gravados no diretório INCLUDE.

# Estrutura Básica do Código C++

- **O ARQUIVO iostream.h**
- O arquivo `iostream.h` é um exemplo de arquivo de inclusão.
- Ele contém declarações necessárias ao uso do objeto `cout` e do operador de inserção `«`.
- Sem essas declarações, o compilador não reconhece `cout` e `«`.
- O arquivo `iostream.h` inclui ainda um objeto para recuperar dados digitados no teclado e também outras definições básicas de I/O (impressão e leitura) necessárias a todos os programas que fizerem uso da saída padrão (vídeo) e da entrada padrão (teclado).

# Estrutura Básica do Código C++

- **EXECUTANDO O PRIMEIRO PROGRAMA**
- O nosso primeiro programa, quando executado, irá imprimir na tela:
- Primeiro Programa
- Se você executar o nosso primeiro programa três vezes seguidas, obterá a seguinte saída:
- Primeiro ProgramaPrimeiro ProgramaPrimeiro Programa

# Estrutura Básica do Código C++

- Note que o objeto `cout` não imprime numa nova linha automaticamente.
- A impressão é colocada na posição atual do cursor.
- Se desejar, você deve inserir um caractere de nova linha explicitamente.
- O caractere de nova linha não pode ser inserido diretamente pelo teclado.
- Por que não usar a tecla [ENTER]?
  - Porque, se você pressionar a tecla [ENTER] em alguma posição na frase que deve ser impressa, o processador de textos que edita o seu programa-fonte abandonará a linha atual e passará para uma nova linha, deixando a linha anterior inacabada, e o objeto `cout` não o tomará como parte da impressão.

# Estrutura Básica do Código C++

- **CÓDIGOS ESPECIAIS**

- Além da tecla [ENTER], vários outros caracteres não podem ser digitados do teclado para dentro do nosso programa.
- Esses caracteres que não podem ser obtidos diretamente do teclado são codificados em C++ por meio da combinação do sinal \ (barra invertida) com outros caracteres.
- A tabela a seguir mostra esses códigos.

CÓDIGOS ESPECIAIS	SIGNIFICADO
\n	Nova linha (CR+LF)
\t	Tab
\b	Retrocesso
\f	Salta página de formulário
\a	Beep — Toca o alto-falante
\r	CR — Cursor para o início da linha
\\	\ — Barra invertida
\0	Null — Zero
\'	' — Aspa simples
\"	" — Aspa dupla
\xdd	Representação hexadecimal

# Estrutura Básica do Código C++

- **IMPRIMINDO NUMA NOVA LINHA**
- Vamos alterar o nosso exemplo para que a frase seja impressa numa nova linha toda vez que o programa for executado.
- Para isso, é necessário inserir um código de nova linha no início da frase a ser impressa.
- Códigos especiais podem ser colocados em qualquer lugar dentro de uma cadeia de caracteres.

```
#include <iostream.h>
main()
{
    cout << "\nPrimeiro programa";
}
```

# Estrutura Básica do Código C++

- **CONSTANTES NUMÉRICAS**
- Uma constante tem valor fixo e inalterável. Números constantes em C++ podem ser escritos nas seguintes bases numéricas:
  - DECIMAL: Escrevemos um número em decimal de forma pura e simples, como a que estamos acostumados.
    - Exemplos: 2345, 50, 88. Observe que números em decimal não podem estar entre aspas.
  - HEXADECIMAL: Os números escritos na base 16 devem ser precedidos de 0x.
    - Exemplo: 0x41, 0x1afb, 0x54c2.

# Estrutura Básica do Código C++

- **CONSTANTES NUMÉRICAS**

- OCTAL: Os números escritos na base 8 devem ser precedidos de um zero na frente.

- Exemplo: 041, 010, 0754.

- CARACTERE: A numeração caractere só está definida para números entre 0 e 255

- Exemplo: '5', 'a', 'w', '\n', '\t'.

- Observação: Note que um zero na frente de um número é avaliado como representação octal.

- Ou seja, escrever 10 não é a mesma coisa que escrever 010. O número 010 é o número 8 em decimal.

# Estrutura Básica do Código C++

- **CADEIA DE CARACTERES CONSTANTE**

- A expressão "Primeiro programa" é um exemplo de cadeia de caracteres constante.
- A linguagem C++ reconhece uma cadeia de caracteres constante quando delimitada por aspas duplas, como mostrado na expressão anterior.

- **ASPAS SIMPLES OU ASPAS DUPLAS**

- Algumas linguagens de programação permitem o uso de aspas simples ou duplas indistintamente.
- Em C++, as aspas simples servem para representar um único caractere ASCII com um valor numérico entre 0 e 255 e as aspas duplas, para representar cadeias de caracteres.

- **IMPRIMINDO OUTROS TIPOS DE DADOS**

- O próximo exemplo mostra como imprimir diferentes tipos de dados utilizando cout.

# Estrutura Básica do Código C++

## ASCII Table

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	\$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(	72	48	110	H	104	68	150	h
9	9	11		41	29	51	)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[	123	7B	173	[
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	\
29	1D	35		61	3D	75	=	93	5D	135	]	125	7D	175	]
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	^
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	_

# Estrutura Básica do Código C++

- Note que utilizamos o operador « repetidamente na instrução.
- Este uso é perfeitamente correto.
- O programa envia primeiramente a cadeia "Venus esta a" para cout, então envia o número 67, e em seguida a cadeia "milhões de milhas", o caractere de nova linha ' \n' e finalmente a cadeia "do sol".

```
#include <iostream.h>
void main()
{
    cout << "Venus esta a " << 67 << "milhoes de milhas"
        << '\n' << "do sol";
}
```

# Estrutura Básica do Código C++

- Vamos agora escrever um programa com mais de uma instrução:

```
#include <iostream.h>
void main()
{
    cout << "A letra " << 'j';
    cout << " pronuncia-se " << "jota" << '.';
}
```

- Observe que 'j' é delimitado por aspas simples, enquanto "jota" é delimitado por aspas duplas. Isto indica ao compilador como diferenciar um caractere de uma cadeia de caracteres.
- Note também que a saída é escrita em duas linhas de programa, o que não constitui duas linhas impressas de texto. Se você desejar imprimir um caractere de nova linha, deve inseri-lo explicitamente.

# Estrutura Básica do Código C++

- VARIÁVEIS
- As variáveis são o aspecto fundamental de qualquer linguagem de computador.
- Uma variável em C++ é um espaço de memória reservado para armazenar um certo tipo de dado e tendo um nome para referenciar o seu conteúdo.
- O espaço de memória ocupado por uma variável pode ser compartilhado por diferentes valores segundo certas circunstâncias.
- Em outras palavras, uma variável é um espaço de memória que pode conter, a cada tempo, valores diferentes.

# Estrutura Básica do Código C++

- Para explicar o uso de variáveis, vamos escrever um programa que cria duas variáveis:
- Se você executar este programa, obterá a seguinte saída: o primeiro número é 44  
O segundo número é 88
- Este programa cria primeiramente a variável num1, imprime o valor atribuído a ela como parte de uma frase, para depois criar a variável num2, atribuir 88 a ela e imprimir seu conteúdo.

```
#include <iostream.h>
void main()
{
    int num1;

    num1=44;
    cout << "\n0 primeiro número é " << num1;

    int num2;

    num2=88;
    cout << "\n0 segundo número é " << num2;
}
```

# Estrutura Básica do Código C++

- **DECLARAÇÕES DE VARIÁVEIS**

- As instruções:
- `int num1;`
- `int num2;`
- são exemplos de declaração de variáveis, isto é, apresentam um tipo `int` e um nome para acessar seu conteúdo.
- Uma declaração de variável é uma instrução para reservar uma quantidade de memória apropriada a fim de armazenar o tipo especificado, nesse caso `int`, e indicar que seu conteúdo será referenciado pelo nome dado a ela.

# Estrutura Básica do Código C++

- Uma declaração de variável consiste no nome de um tipo, seguido do nome da variável, seguido de ponto-e-vírgula.
- Em C++, todas as variáveis devem ser declaradas.
- As variáveis devem ser declaradas antes de serem usadas no programa.
- Uma variável pode ser declarada em qualquer lugar do programa.
- C++ não funciona se você não declarar suas variáveis.
- As variáveis podem ser declaradas em qualquer lugar do programa.
- Se você tiver mais de uma variável do mesmo tipo, poderá declará-las de uma única vez, separando seus nomes por vírgulas:
- `int aviao, foguete, helicóptero;`

# Estrutura Básica do Código C++

- **TIPOS DE VARIÁVEIS**
- O tipo de uma variável informa a quantidade de memória, em bytes, que a variável ocupará e a forma como um valor deverá ser armazenado e interpretado.
- Tipo de variável diz respeito ao tamanho de memória e à forma de armazenamento.
- Em C++, existem cinco tipos básicos de variáveis.
- Nos computadores de ambiente Windows e Linux, a tabela seguinte é válida:

# Estrutura Básica do Código C++

- Com exceção de void, os tipos de dados básicos podem ser acompanhados por modificadores na declaração de variáveis.
- Os modificadores de tipo oferecidos por C++ são: long, short e unsigned;

TIPO	BIT	BYTES	ESCALA
char	8	1	-128 a 127
int	16	2	-32768 a 32767
float	32	4	3.4E-38 a 3.4E+38
double	64	8	1.7E-308 a 1.7E+308
void	0	0	nenhum valor

# Estrutura Básica do Código C++

- Um modificador de tipo pode ser utilizado sem que seja especificado o tipo da variável.
- Quando isso é feito, o compilador assume, por default, que o tipo é int.
- Os modificadores podem ser utilizados com os tipos e as escalas descritos pela tabela a seguir:

TIPO	BIT	BYTES	ESCALA
unsigned char	8	1	0 a 255
unsigned	16	2	0 a 65535
short	16	2	-32768 a 32767
long	32	4	-2147483648 a 2147483647
unsigned long	32	4	0 a 4294967295
long double	80	10	3.4E-4932 a 1.1E+4932

# Estrutura Básica do Código C++

- O tipo `int` tem sempre o tamanho da palavra da máquina, isto é, em computadores de 16 bits ele terá 16 bits de tamanho.
- O tipo `short` tem tamanho diferente do tipo `int` em outros computadores, geralmente a metade do tamanho de um `int`.
- No geral, o tipo `short` é idêntico ao tipo `int` e, por esse motivo, é pouco usado.

# Bibliografia

- MIZRAHI1, Victorine Viviane. Treinamento Em Linguagem C++ - Modulo 1. 2ª Edição. Editora: Pearson Universidades. Ano: 2005. ISBN: 8576050455.
- VIEIRA, Lucas S. Introdução a Programação em C++. Diamantina - MG. Creative Commons – 2019.
- PEREIRA, Silvio do Lago. Linguagem C++. São Paulo: FATEC, 1999.
- MIZRAHI2, Victorine Viviane. Treinamento Em Linguagem C++ - Modulo 2. 2ª Edição. Editora: Pearson Universidades. Ano: 2005. ISBN: 8576050463.