



Linguagem de Programação I

Neilor Tonin

2017

Introdução

```
//Meu primeiro programa

#include <iostream>

using namespace std;

int main() {
    string nome = "Ola mundo!";
    cout << nome << endl;
    return 0;
}
```

Ola mundo

Uso de Namespaces

```
// namespaces
#include <iostream>

using namespace std;

namespace first {
    int var = 5;
}

namespace second {
    double var = 3.1416;
}

int main () {
    int var = 2;
    cout << var << endl;
    cout << first::var << endl;
    cout << second::var << endl;
    return 0;
}
```

```
2
5
3.1416
```


Bibliotecas

- C e C++

Include no Ansi C:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <math.h>
```

Include no C++

```
#include <cstdio>
#include <cstdlib>
#include <cstring>
#include <ctime>
#include <cmath>
```

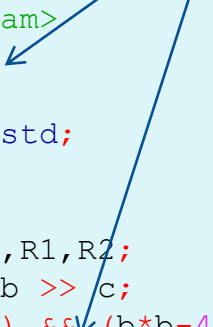
Criação de Bibliotecas

- Podemos utilizar qualquer biblioteca disponível em C ou C++

```
#include <iostream>
#include <cmath>

using namespace std;

int main() {
    double a,b,c,R1,R2;
    cin >> a >> b >> c;
    if ( (a !=0 ) && (b*b-4*a*c>=0) ) {
        R1= (-b + sqrt(b*b-4*a*c) ) / (2*a);
        R2= (-b - sqrt(b*b-4*a*c) ) / (2*a);
        cout << "Raiz 1=" << R1 << endl;
        cout << "Raiz 2=" << R2 << endl;
    } else {
        cout << "Valores invalidos" << endl;
    }
    return 0;
}
```



```
2 6 1
Raiz 1=-0.177124
Raiz 2=-2.82288
```

Criação de Bibliotecas

- Podemos agrupar funções de um tipo específico ou funções criadas pelo usuário

```
#include <iostream>
#include "minhalib.h"

using namespace std;

int main() {
    double valor;
    cout << "Digite um valor: ";
    cin >> valor;
    // funcao que esta no minhalib.h
    mostradobro(valor);
    mostratriplo (valor);
    return 0;
}
```

```
Digite um valor: 2
4
6
```


Criação de Bibliotecas

- Podemos agrupar funções de um tipo específico ou funções criadas pelo usuário

```
#include <iostream>
#include "minhalib.h"

using namespace std;

int main() {
    double valor;
    cout << "Digite um valor: ";
    cin >> valor;
    // funcao que esta no minhalib.h
    mostradobro(valor);
    mostratriplo (valor);
    mostra_ao_cubo (valor);
    cout << "Fatorial de " << valor << " = "
         << fatorial(valor) << endl;
    return 0;
}
```

Digite um valor:
3
O dobro de 3 = 6
O triplo de 3 = 9
3 ao cubo = 27
Fatorial de 3 = 6

Uso de Defines

Em substituição a funções simples:

```
#include <iostream>
#define triplo(x) (x*3)
#define mensagem "Digite um valor: "

using namespace std;

int main(void){
    int a;
    cout << mensagem;
    cin >>a;
    cout << "Triplo de " << a << " = " << triplo(a) << "\n";
    return (0);
}
```

```
Digite um valor: 4
Triplo de 4 = 12
```


Uso de Defines

Exercício 1 : faça funcionar o seguinte programa

```
#include <iostream>
#define triplo(x) (x*3)
#define mensagem "Digite um valor: "

using namespace std;

int main(void){
    int a;
    escrever mensagem;
    cin >>a;
    escrever "Triplo de "<< a <<
        " = " << triplo(a) <<"\n";
    return (0);
}
```

```
Digite um valor: 4
Triplo de 4 = 12
```

Uso de Defines

Exercício 2 : faça funcionar o seguinte programa

```
#include <iostream>
#define triplo(x) (x*3)
#define mensagem "Digite um valor: "

using namespace std;

programa {
    int a;
    escrever mensagem;
    ler a;
    escrever "O triplo de " << a <<
        " eh " << triplo(a) << "\n";
    fim;
}
```

Digite um valor: 4
Triplo de 4 = 12

Exercício 3 : faça funcionar o seguinte programa

```
int main () {
    int a=3,b=6, i;
    double x=7.3;
    cout << "a: " <<a<< " b: " <<b<< " x: " << x << endl;
    cout << "Dobro de a: " << dobro(a) << " " << endl;
    cout << "Quadrado de a: " << quadrado(a) << " " << "\n";
    fori(i, a, b) { /// for (i =a; i<=b; i++)
        cout << i << " ";
    }
    cout << endl;
    fori(i, 6, 10) { /// for (i =6; i<=10; i++)
        cout << i << " ";
    }
    cout << endl;
    ford(i, TAM-15, 0) {
        cout << i << " ";
    }
    cout << endl;
    cout << potencia(2,3) << endl;
    cout << absoluto(-6.5) << endl;
    cout << fixed << setprecision(12);
    cout << raiz(2) << endl;
    return(0);
}
```

```
a: 3 b: 6 x: 7.3
Dobro de a: 6
Quadrado de a: 9
3 4 5 6
6 7 8 9 10
5 4 3 2 1 0
8
6.5
1.414213562373
```


Tipos de datos básicos

Name	Description	Size*	Range*
char	Character or small integer.	1byte	signed: -128 to 127 unsigned: 0 to 255
short int (short)	Short Integer.	2bytes	signed: -32768 to 32767 unsigned: 0 to 65535
int	Integer.	4bytes	signed: -2147483648 to 2147483647 unsigned: 0 to 4294967295
long int (long)	Long integer.	4bytes	signed: -2147483648 to 2147483647 unsigned: 0 to 4294967295
bool	Boolean value. It can take one of two values: true or false.	1byte	true or false
float	Floating point number.	4bytes	+/- 3.4e +/- 38 (~7 digits)
double	Double precision floating point number.	8bytes	+/- 1.7e +/- 308 (~15 digits)
long double	Long double precision floating point number.	8bytes	+/- 1.7e +/- 308 (~15 digits)
wchar_t	Wide character.	2 or 4 bytes	1 wide character

Tipos de dados primitivos

a) Limite das variáveis

```
#include <iostream>

using namespace std;

int main() {
    int c = 3123456789;
    cout << c << endl;

    unsigned int cc = 3123456789;
    cout << cc << endl;

    cc = 4294967295;
    cout << cc << endl;

    cc ++;
    cout << cc << endl;

    return 0;
}
```

```
-1171510507
3123456789
4294967295
0
```

```
Process returned 0 (0x0)
Press any key to continue.
```

b) Outras variáveis e deslocamento de bits

```
int main() {  
    long long int d = 1234567890123456789;  
    cout << d << endl;  
  
    unsigned long long int dd = 12345678901234567890;  
    cout << dd << endl;  
  
    int e = ( 3 << 4); ///desloca de bits p/ esquerda  
    cout << e << endl;  
  
    /// Um inteiro ocupa 4 bytes (32 bits)  
    e = (1<<31) - 1;    ///2147483647;  
    cout << e << endl;  
  
    unsigned int f=((1<<32) - 1);    ///4294967295;  
    cout << f << endl;  
  
    e=((1<<32));  
    cout << e << endl;  
  
    unsigned long long g=((1<<64) - 1);  
    cout << g << endl << endl;  
  
    return 0;  
}
```

```
1234567890123456789  
12345678901234567890  
48  
2147483647  
4294967295  
0  
18446744073709551615
```

Process returned 0
(0x0)
Press any key to
continue.

c) Tipos de variáveis. Continuação...

```
int main(void) {  
    char a=65;    ///65=A 66=B 97=a  
    cout << a << endl;  
    cout << (int) a << endl;  
    cout << (char) a << endl << endl;  
  
    short int b=-32768; //vai até 32767  
    cout << b << endl;  
  
    b = b-1;  
    cout << b << endl;  
    unsigned short int bb=65535;  
    cout << bb << endl;  
  
    bb = bb+1;  
    cout << bb << endl;  
  
    return 0;  
}
```

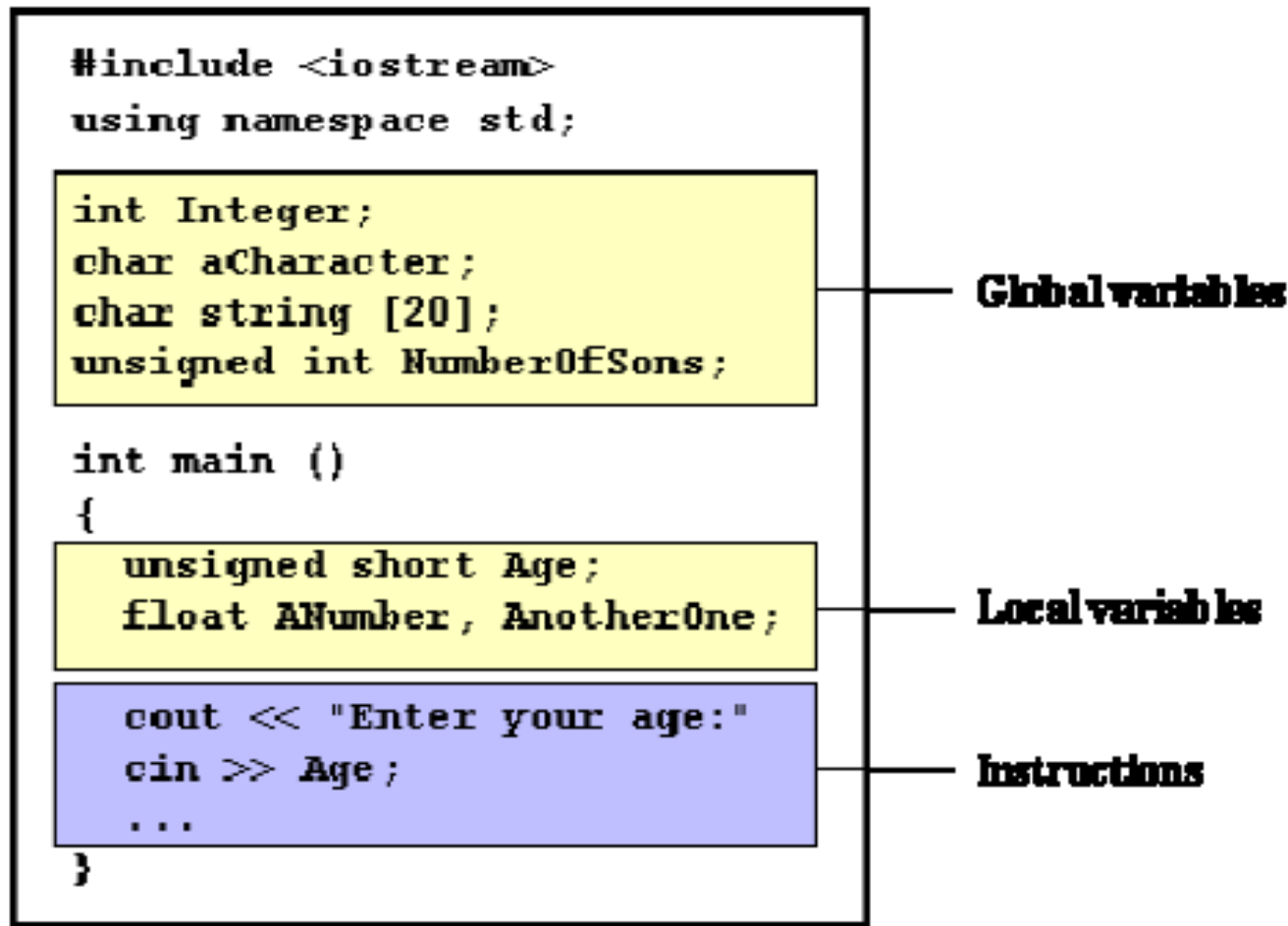
A
65
A

-32768
32767
65535
0

Process returned 0 (0x0)
Press any key to continue.

Escopo de Variáveis

- Conforme livro, temos variáveis globais, locais e instruções:



Escopo de Variáveis

```

/*****
Aula exemplo de escopo de variáveis
Autor: Neilor
Data: 02/08/2017
*****/

#include <iostream>

using namespace std;

int i=23;

void mangojata() {
    cout << "i global: " << i << endl;
}

int main() {
    int i = 10;
    for (int i=1; i<=5; i++) {
        cout << "dentro do for: " << i << endl;
    }
    cout << "dentro do main(): " << i << endl;
    mangojata();
    return 0;
}

```

???

Escopo de Variáveis

```
/******  
Aula exemplo de escopo de variáveis  
******/  
#include <iostream>  
  
using namespace std;  
  
int i=23; ///variavel global  
  
void mangojata() {  
    cout << "i global: " << i << endl;  
}  
void mangojata_mancuda(int i) {  
    cout << "i parametro: " << i << endl;  
}  
  
int main() {  
    int i = 10; /// variavel local (todo o programa main)  
    for (int i=1; i<=5; i++) { /// variavel local do laço for  
        cout << "dentro do for: " << i << endl;  
    }  
    cout << "dentro do main(): " << i << endl;  
    mangojata();  
    mangojata_mancuda(40);  
    return 0;  
}
```

???