Estrutura
Variáveis
Entrada/Saída
Atribuição

INF110 – Programação I

Prof. Alcione/André Gustavo
DPI/UFV – 2020/1

UFV

+ A linguagem C/C++

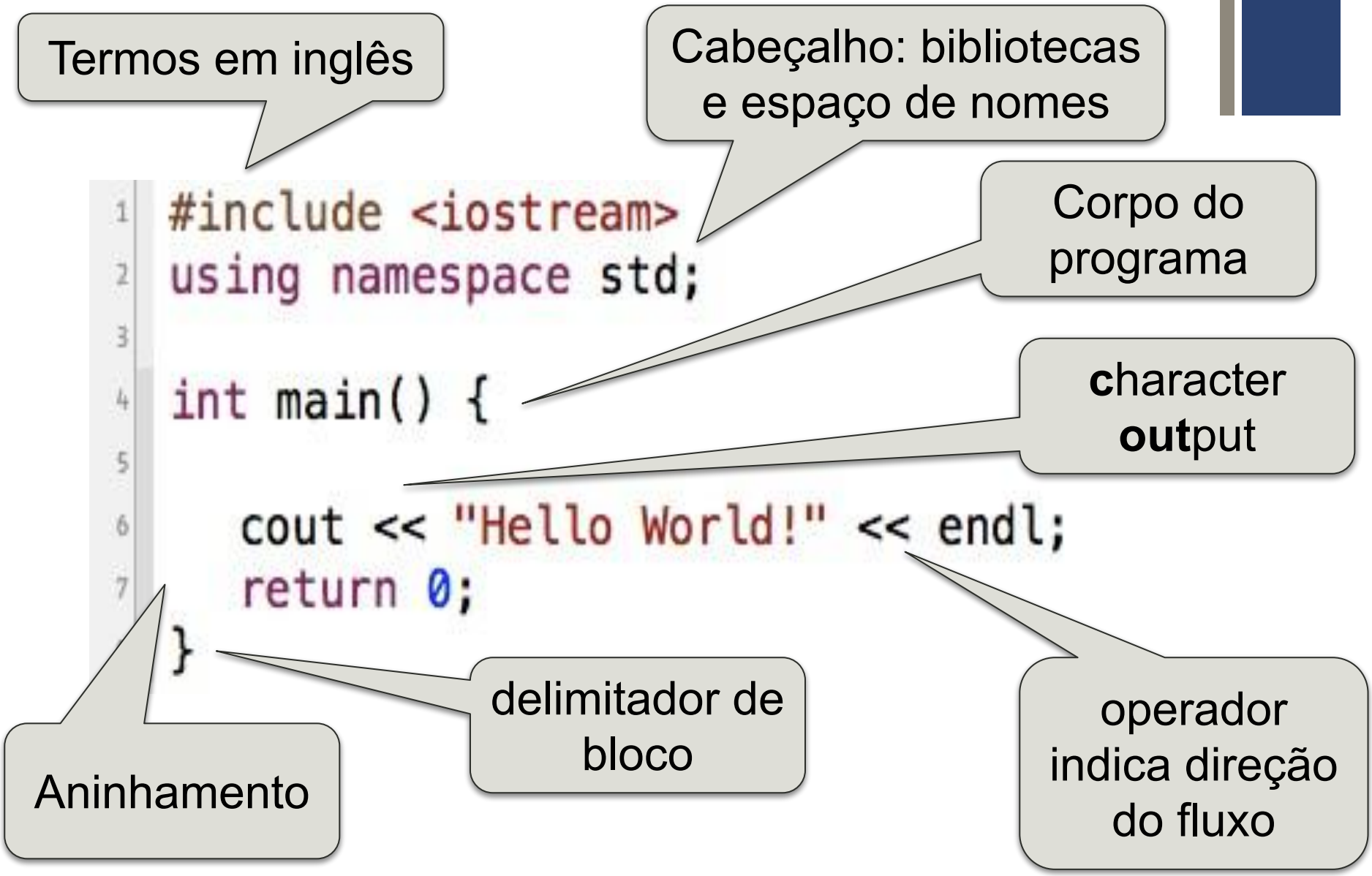
- C é uma linguagem de programação de computadores desenvolvida em 1972 por Dennis M. Ritchie, no Bell Telephone Laboratories, para desenvolver o sistema operacional UNIX. C é uma linguagem de programação simples e orientada para a estrutura.

- A linguagem C ++ é desenvolvida por Bjarne Stroustrup no ano de 1979 no bell laboratory nos EUA, C ++ é uma linguagem de programação simples e orientada a objetos.

+ Primeiro programa em C/C++

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5
6     cout << "Hello World!" << endl;
7     return 0;
8  }
```

+ Primeiro programa em C/C++



Primeiro programa em C/C++

- O aninhamento (identação) permite entender que comando ocorre no escopo de outro comando
  - Aumenta a legibilidade
- Os delimitadores definem o escopo de um comando
- Os “;” indicam o fim de um comando

Primeiro programa em C/C++

- Executando

Primeiro programa em C/C++

Instalando no g++ no Windows

1. Baixar o arquivo MinGW (https://osdn.net/projects/mingw/downloads/68260/mingw-get-setup.exe)
2. Descompactar o arquivo em alguma pasta, sem espaço no nome.(por exemplo ***C:\MinGW***)
3. Acrescentar essa pasta na variável de ambiente PATH do windows:
- Entrar em **Control Panel -> System -> Advanced System Settings -> Environment Variables**
- Clicar em **PATH** em **User Variables** ou **System Variables**
- Acrescentar a pasta **bin** do local de instalação do MinGW ao final. Por exemplo: ...;***C:\MinGW***;
(onde ... é o que já tinha antes no valor de PATH).
4. Pronto!

Primeiro programa em C/C++

- Alternativas online

C/C++ online

http://cpp.sh/

https://www.onlinegdb.com/

https://repl.it/languages/

https://www.jdoodle.com/c-online-compiler/

https://ideone.com/

Outro programa em C/C++

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

int main() {

    int prova1, prova2, prova3, notafinal;

//Ler nota da prova 1, prova 2, prova 3
    cout << "Digite sua nota na prova 1: ";
    cin >> prova1;
    cout << "Digite sua nota na prova 2: ";
    cin >> prova2;
    cout << "Digite sua nota na prova 3: ";
    cin >> prova3;

//Calcular a nota final: somar as 3 notas
    notafinal = prova1 + prova2 + prova3 ;

//Escrever a nota final
    cout << "Sua nota final foi  ";
    cout << notafinal << endl ;

    return 0;
}
```

Outro programa em C/C++

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

int main() {

    int prova1, prova2, prova3, notafinal;

//Ler nota da prova 1, prova 2, prova 3
    cout << "Digite sua nota na prova 1: ";
    cin >> prova1;
    cout << "Digite sua nota na prova 2: ";
    cin >> prova2;
    cout << "Digite sua nota na prova 3: ";
    cin >> prova3;

//Calcular a nota final: somar as 3 notas
    notafinal = prova1 + prova2 + prova3 ;

//Escrever a nota final
    cout << "Sua nota final foi  ";
    cout << notafinal << endl ;

    return 0;
}
```

Entrada

Processamento

Saída

Outro programa em C/C++

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5
6      int prova1, prova2, prova3, notafinal;
7
8  //Ler nota da prova 1, prova 2, prova 3
9      cout << "Digite sua nota na prova 1: ";
10     cin >> prova1;
11     cout << "Digite sua nota na prova 2: ";
12     cin >> prova2;
13     cout << "Digite sua nota na prova 3: ";
14     cin >> prova3;
15
16 //Calcular a nota final: somar as 3 notas
17     notafinal = prova1 + prova2 + prova3 ;
18
19 //Escrever a nota final
20     cout << "Sua nota final foi  ";
21     cout << notafinal << endl ;
22
23     return 0;
24 }
```

Declaração de variáveis

Entrada

Processamento

Saída

Variável

- “Local na memória” com endereço, nome, tipo e valor
- Nome (identificador)
  - Em C++ somente letras (A-Z, a-z), dígitos (0-9) e sublinhado (_)
  - Não pode começar com dígito
- Tipo
  - Exemplo em C++: `int`
- Valor
  - Um único valor (se colocar outro, apaga o antigo)
- Endereço (localização na memória)
  - Programador não precisa saber, pode acessar pelo identificador

Variável

- Declaração
  - Informar o tipo e o nome
  - O computador se encarrega de definir o endereço
  - Exemplo em C++

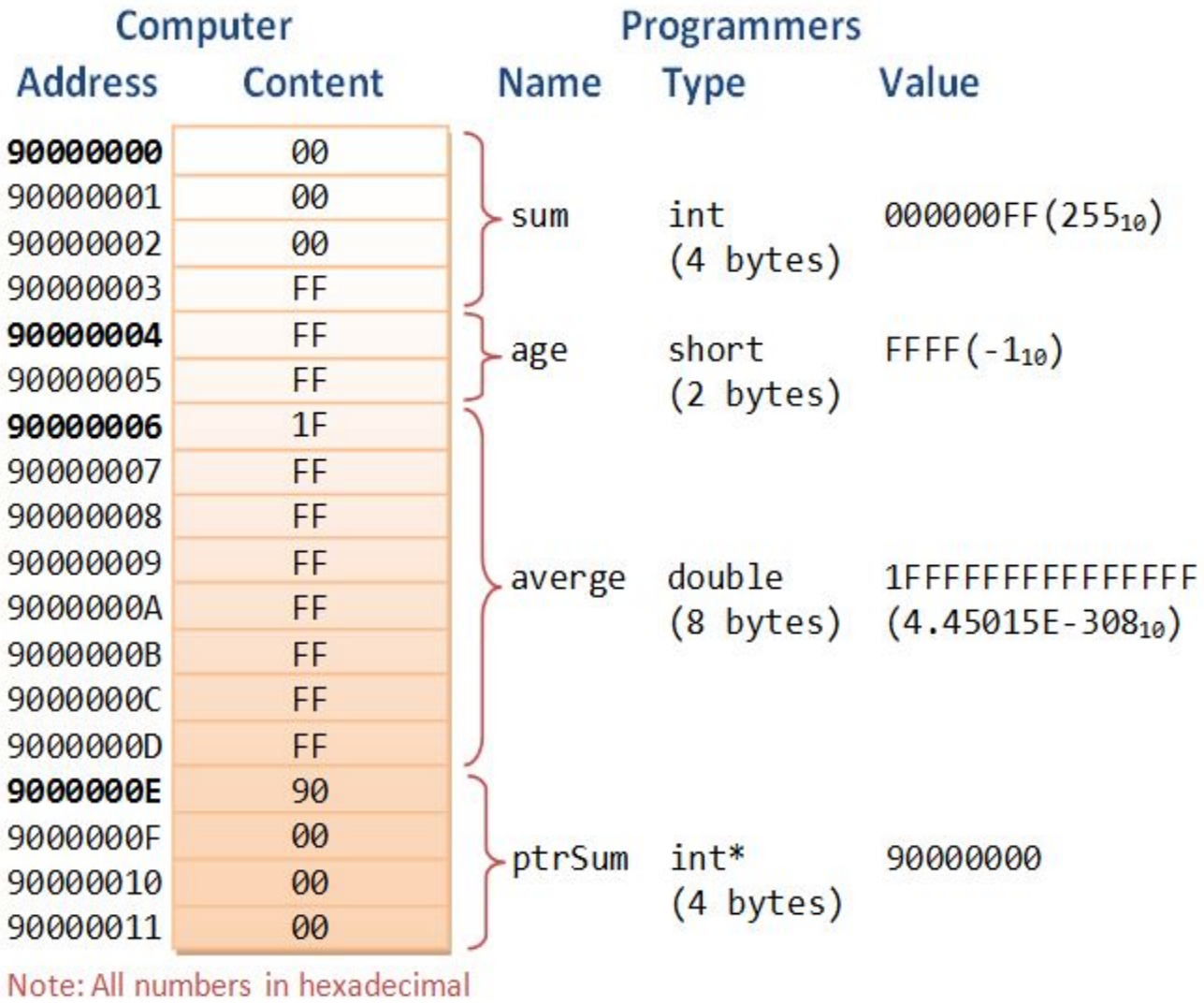
```
int prova1;
int prova2;
int prova3;
int notafinal;
```

ou

```
int prova1, prova2, prova3, notafinal;
```

- Valor
  - Inicialmente um “lixo”
  - Pode ser atualizado por leitura de entrada ou atribuição

Variável



https://www3.ntu.edu.sg/home/ehchua/programming/cpp/cp4_PointerReference.html

Entrada – cin

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

- Sintaxe
  - `cin >> variável;`
- Funcionamento
  - O programa interrompe a execução e aguarda entrada de dados
  - Só avança quando for digitado um dado e pressionado ‘enter’
- Pode ser lida uma lista de dados
  - `cin >> variávelA >> variávelB >> variávelC;`
- Funcionamento
  - Os dados são armazenados nas variáveis na ordem digitada
  - Só avança após serem lidos todos os dados

Saída – cout

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

- Sintaxe
  - `cout << **expressão;**`
- Funcionamento
  - Avalia a expressão e escreve seu valor, de acordo com seu tipo
  - Se for uma variável, escreve seu valor
  - Se for uma expressão aritmética, avaliar e escreve o resultado
  - Se for um texto entre aspas (“ ”), escreve o texto
- Pode ser escrita uma lista de expressões
  - `cout << **expressãoA** << **expressãoB** << **expressãoC;**`

Saída – endl

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

- Sintaxe
  - `cout << endl;`
- Funcionamento
  - Usado na saída com
  - Encerra uma linha (o próximo dado sairá na linha seguinte)
- Pode ser usado dentro de uma lista de expressões
  - `cout << **expressãoA** << endl << **expressãoB** << endl;`

Saída – '\n'

- Caracter especial, indica mudança de linha
  - Mesmo efeito que `endl`, mas usado dentro de texto entre " "
- Exemplo:

```
cout << "==UFV==\nBem-vindo!\nDigite sua matricula: ";
```

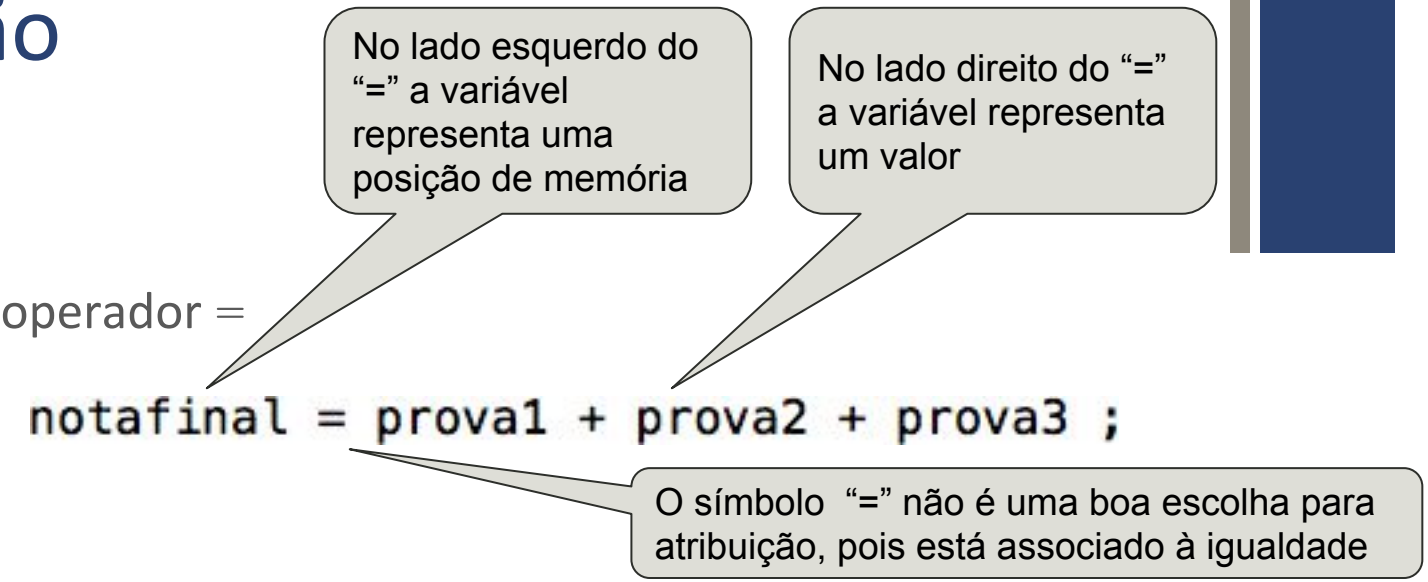
- Resultado:

```
==UFV==
Bem-vindo!
Digite sua matricula:
```

- Compare com a versão usando `endl`:

```
cout << "==UFV==" << endl << "Bem-vindo!" << endl << "Digite sua matricula: ";
```

Atribuição



- É feita com o operador =
- Exemplo:
- Resultado:
  - a expressão do lado direito do '=' é avaliada e seu valor é atribuído à variável que está do lado esquerdo do '='
- Observações:
  - Do lado esquerdo deve haver uma e somente uma variável
  - Não cria "vínculo", a atribuição é feita quando e somente quando o comando é executado
  - Substitui o valor antigo da variável. Se forem atribuídos valores várias vezes, permanecerá apenas o último atribuído.

Operadores aritméticos

- + adição
- - subtração
- * multiplicação
- / divisão
- % resto (da divisão), chamado “mod”

Operadores aritméticos

- Precedência como na matemática
  - Primeiro: * / % da esquerda para direita
  - Depois: + − da esquerda para direita
- Exemplo:

```
//Calcular a nota final: media das 3 notas
    notafinal = prova1 + prova2 + prova3 / 3 ;
```

Operadores aritméticos

- Precedência como na matemática
  - Primeiro: `*` `/` `%` da esquerda para direita
  - Depois: `+` `-` da esquerda para direita
- Exemplo:

```
//Calcular a nota final: media das 3 notas  ERRADO!!
    notafinal = prova1 + prova2 + prova3 / 3 ;
```

  - Primeiro vai dividir somente a nota da `prova3` por 3 e depois somar

Operadores aritméticos

- Precedência como na matemática
  - Primeiro: `*` `/` `%` da esquerda para direita
  - Depois: `+` `-` da esquerda para direita
- Exemplo:

```
//Calcular a nota final: media das 3 notas  ERRADO!!
    notafinal = prova1 + prova2 + prova3 / 3 ;
```

  - Primeiro vai dividir somente a nota da `prova3` por 3 e depois somar
- Uso de ( )'s:

```
//Calcular a nota final: media das 3 notas
    notafinal = (prova1 + prova2 + prova3) / 3 ;
```

  - Agora sim, primeiro dentro dos ( )'s, ou seja, soma, depois divide

Operadores aritméticos

- Precedência como na matemática
  - Primeiro: `*` `/` `%` da esquerda para direita
  - Depois: `+` `-` da esquerda para direita
- Exemplo:

```
//Calcular a nota final: media das 3 notas  ERRADO!!
    notafinal = prova1 + prova2 + prova3 / 3 ;
```

  - Primeiro vai dividir somente a nota da `prova3` por 3 e depois somar
- Uso de ( )'s:

```
//Calcular a nota final: media das 3 notas
    notafinal = (prova1 + prova2 + prova3) / 3 ;
```

  - Agora sim, primeiro dentro dos ( )'s, ou seja, soma, depois divide

Obs.: não se usa [ ] *nem* { }*, mas* ( ) *dentro de* ( )*, quando necessário*

Terminologia

Variáveis - Em linguagens imperativas (C/C++, Java, Python, etc), variáveis representam posições de memória.

Expressões - São porções do código de linguagem de programação que geram valor. Por exemplo, `3+5` é uma expressão que gera o valor `8`.

Comandos - São porções do código de linguagem de programação que realizam uma ação. Por exemplo, atribuição é um comando pois `a=3 + 5` realiza a ação de inserir o valor 8 na posição de memória representada por `a`.

Continua em:

Variáveis e Operadores aritméticos