

ALGORITMOS E ESTRUTURAS DE DADOS I



UFOP



decom

departamento
de computação

INTRODUÇÃO AO C

Prof.: Edwaldo Soares Rodrigues

E-mail: edwaldo.rodrigues@uemg.br

INTRODUÇÃO AO C



UFOP



decom

departamento
de computação

- Uma das principais linguagens já criadas;
- Utilizada na criação do sistema operacional Unix;



CARACTERÍSTICAS DO C



UFOP



decom

departamento
de computação

- Considerada de “baixo/médio” nível
- Uma das linguagens comerciais mais populares
- Muito utilizada na programação de dispositivos de hardware;

COMPILAÇÃO



UFOP



decom

departamento
de computação



COMPILAÇÃO



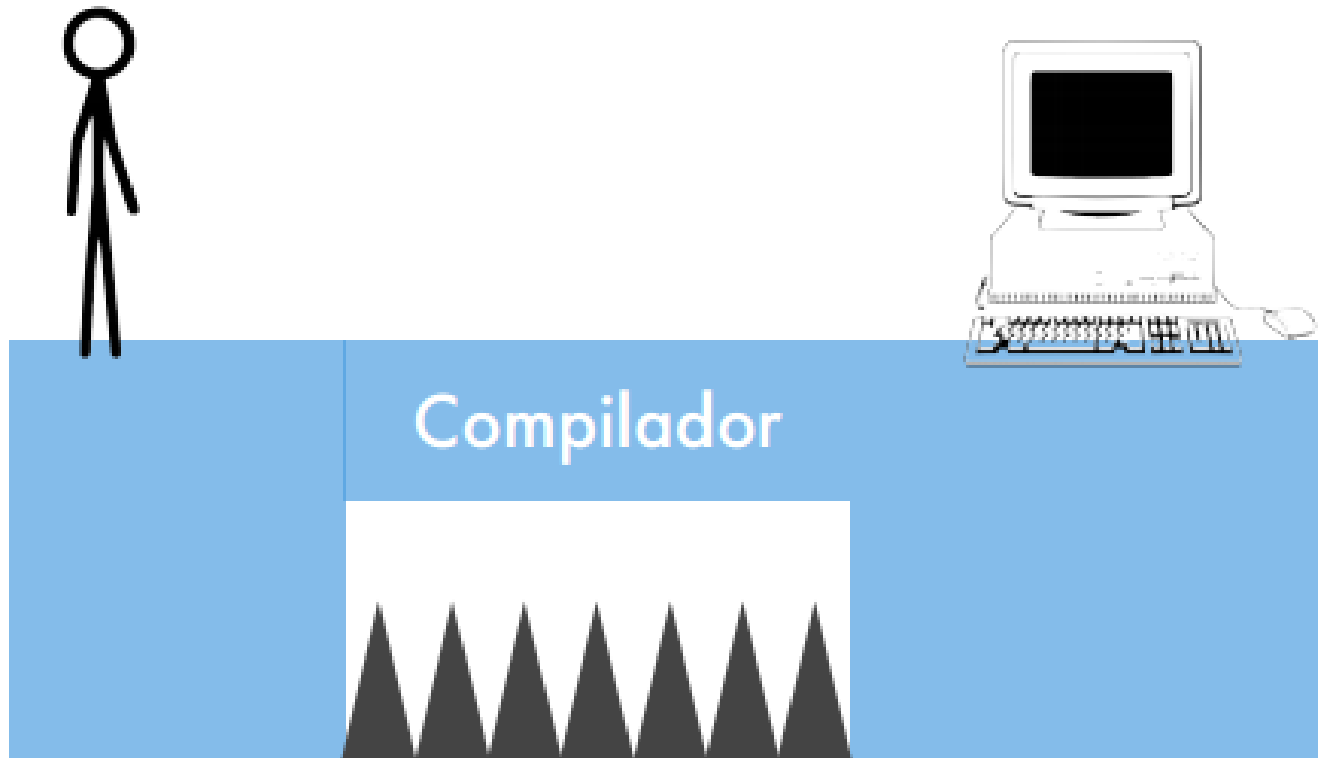
UFOP



decom

departamento
de computação

- O compilador é a ferramenta que nos serve como uma ponte para estabelecer esta comunicação



COMPILAÇÃO



UFOP



decom

departamento
de computação

```
public class TopClientExample
{
    public static void Main()
    {
        byte[] data = new byte[1024]; string input; int offset;
        TopClient server;
        try
        {
            server = new TopClient("...", 8080);
        }
        catch (SocketException e)
        {
            Console.WriteLine("Failed to connect to server");
            return;
        }
        NetworkStream ns = server.GetStream();
        int recv = ns.Read(data, 0, data.Length);
        string data = Encoding.ASCII.GetString(data, 0, recv);
        Console.WriteLine(data);
        while (true)
        {
            input = Console.ReadLine();
            if (input == "exit") break;
            ns.Write(data, 0, data.Length);
        }
    }
}
```

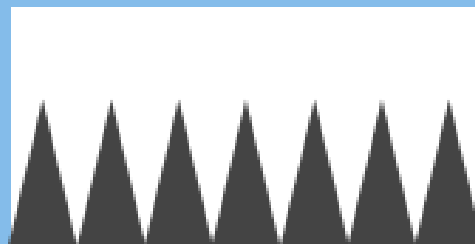


Primeiramente
escreveremos nossa
mensagem para o
computador em uma
linguagem de
programação de
nossa preferência.



Compilador

Código em
linguagem de
programação



COMPILAÇÃO



UFOP



decom

departamento
de computação

```
public class TopClientExample
{
    public static void Main()
    {
        byte[] data = new byte[1024];
        TopClient server;
        try
        {
            server = new TopClient("localhost", 8080);
        }
        catch (IOException e)
        {
            Console.WriteLine("Failed to connect to server");
            return;
        }
        NetworkStream ns = server.GetStream();
        int recv = ns.Read(data, 0, data.Length);
        stringData = Encoding.ASCII.GetString(data, 0, recv);
        Console.WriteLine(stringData);
        while (true)
        {
            input = Console.ReadLine();
            if (input == "exit") break;
            Console.WriteLine("Received: " + input);
            Console.WriteLine("Sending: " + input);
            ns.Write(input, 0, input.Length);
        }
    }
}
```



Compilador recebe o
código



Compilador



COMPILAÇÃO

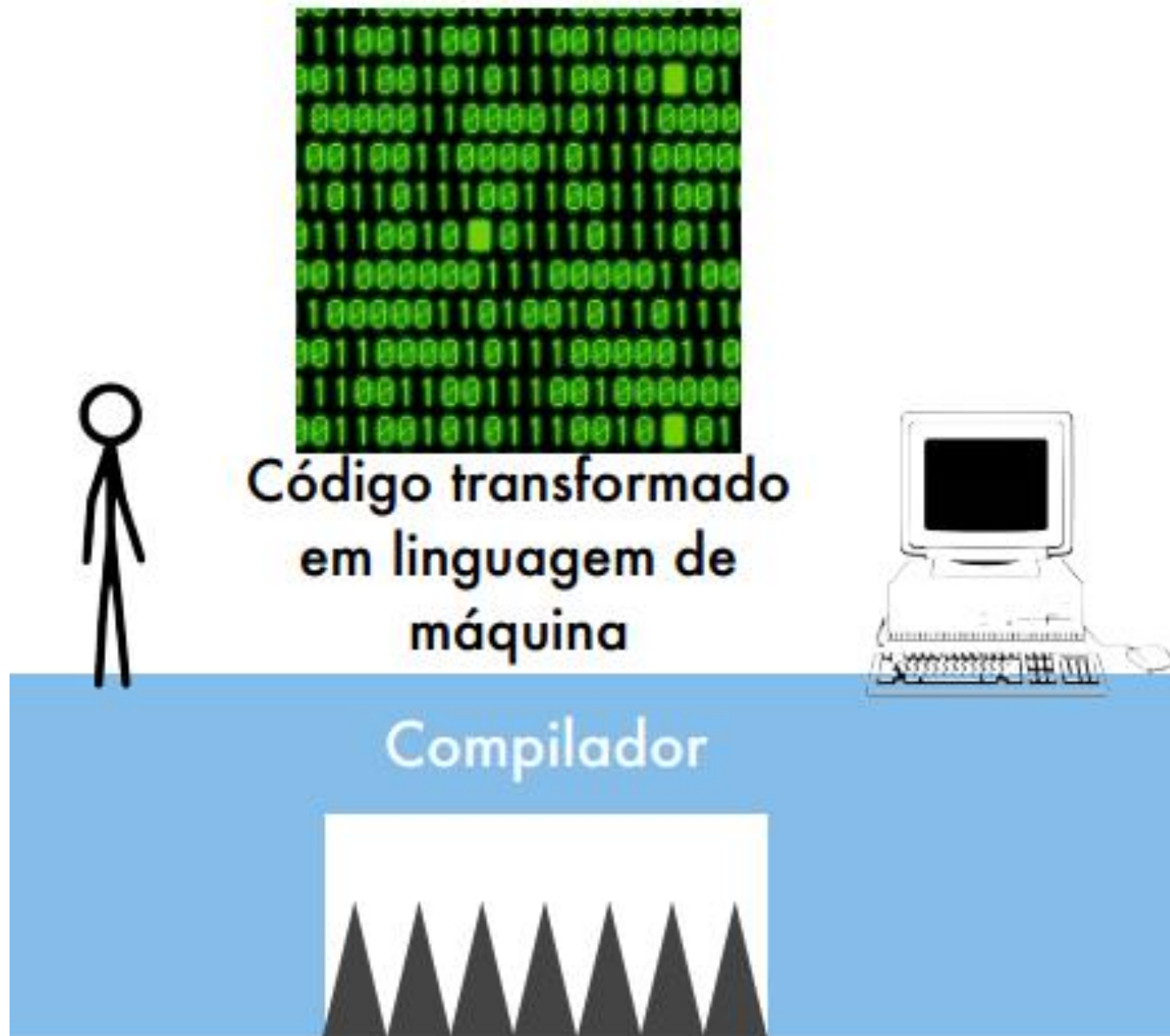


UFOP



decom

departamento
de computação



COMPILAÇÃO

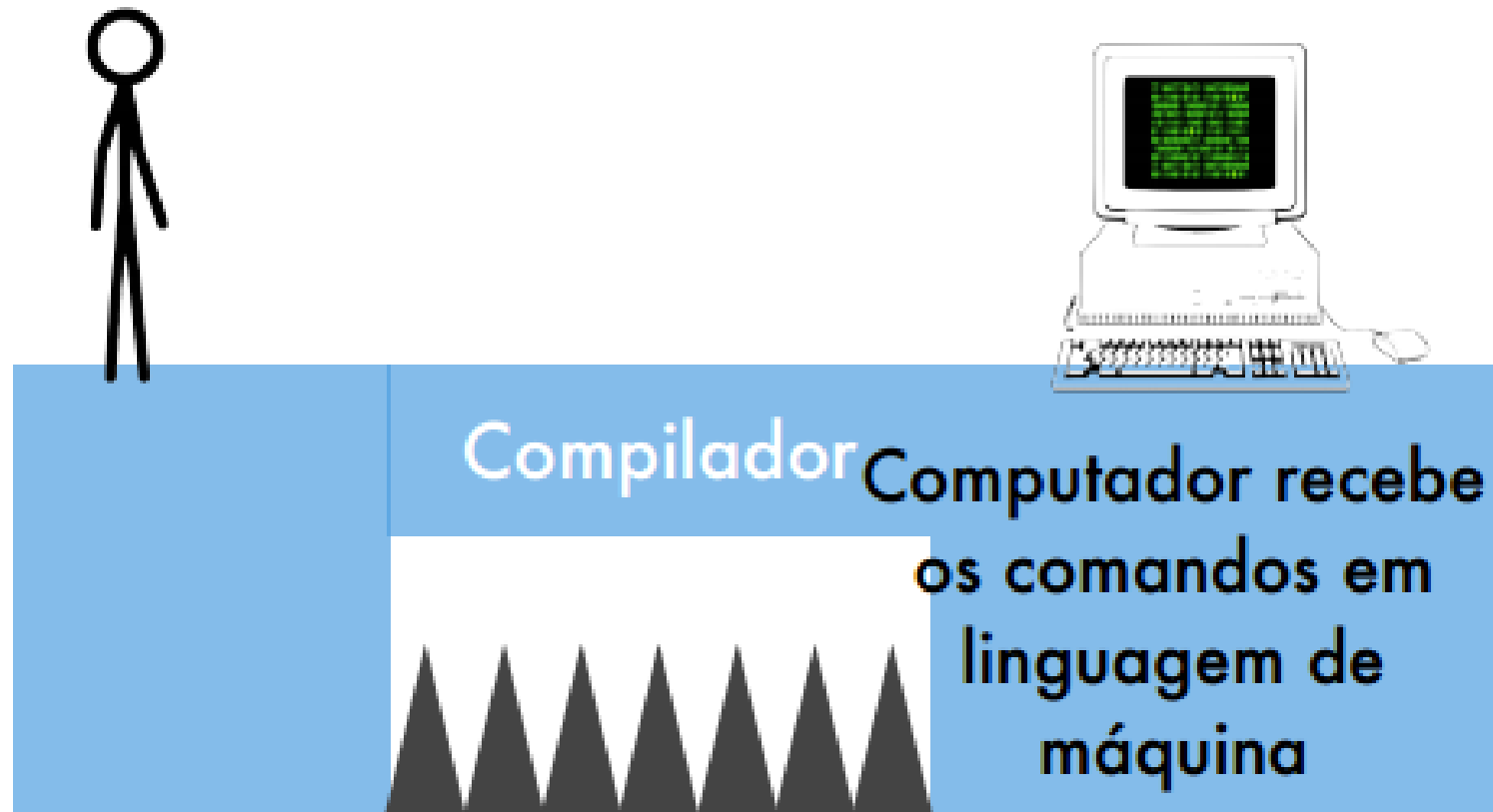


UFOP



decom

departamento
de computação



TIPOS DE DADOS E SEUS TAMANHOS



UFOP



decom

departamento
de computação

0100110101



TIPOS DE DADOS E SEUS TAMANHOS



UFOP



decom

departamento
de computação

0100110101

a? 4? F? 3.5?
?????



TIPOS DE DADOS E SEUS TAMANHOS



UFOP



decom

departamento
de computação

Tipo lógico	bool	V ou F
Números inteiros	short < int < long	... -3, -2, -1, 0, 1, 2, 3...
Números reais	float < double	3.32, 4.78, 7.24, -3.14, 0.01
Caracteres	char	a, b, c, d, e, f ... #, \$, %, \ ...

PRIMEIRO PROGRAMA EM C



UFOP



decom

departamento
de computação

```
#include <stdio.h>

int main(){
    printf("Olá Mundo!");
    return 0;
}
```



PRIMEIRO PROGRAMA EM C



UFOP



decom

departamento
de computação

```
#include <stdio.h>
```

```
int main(){  
    printf("Olá Mundo!");  
    return 0;  
}
```

- O comando em vermelho inclui a biblioteca com recursos para fluxo de entrada e saída padrão (std in/out). As entradas serão feitas pelo teclado, já as saídas pelo monitor

PRIMEIRO PROGRAMA EM C



UFOP



departamento
de computação

```
#include <stdio.h>

int main(){
    printf("Olá Mundo!");
    return 0;
}
```

- Aqui começa a função principal (*main*) de nosso programa. As chaves { e } indicam onde começa e termina a função

PRIMEIRO PROGRAMA EM C



UFOP



decom

departamento
de computação

```
#include <stdio.h>

int main(){
    printf("Olá, Mundo!");
    return 0;
}
```

- Dizemos que *printf* é um objeto que faz saída de dados para o monitor. Toda vez que desejamos emitir uma mensagem na tela do computador faremos uso deste comando;

PRIMEIRO PROGRAMA EM C++



UFOP



departamento
de computação

```
#include <stdio.h>

int main(){
    printf("Olá Mundo!");
    return 0;
}
```

- Com este comando é impresso na tela a mensagem **Olá Mundo!**;



PRIMEIRO PROGRAMA EM C++



UFOP



departamento
de computação

```
#include <stdio.h>

int main(){
    printf("Olá Mundo!");
    return 0;
}
```

- As aspas ao redor da expressão “Olá Mundo!” indicam que é um conjunto de *char_s* e não uma palavra reservada da linguagem;

PRIMEIRO PROGRAMA EM C



UFOP



departamento
de computação

```
#include <stdio.h>

int main(){
    printf("Olá, Mundo!\n");
    return 0;
}
```

- Além da frase, uma quebra de linha após a frase é enviada para *printf*. A quebra de linha é representada por meio do comando **\n**;

PRIMEIRO PROGRAMA EM C



UFOP



decom

departamento
de computação

```
#include <stdio.h>

int main(){
    printf("Olá, Mundo!\n");
    return 0;
}
```

- O comando **return** indica que a função retorna dali encerrando o programa. Ela não só retorna, mas retorna um valor do tipo **int**;

PRIMEIRO PROGRAMA EM C



UFOP



decom

departamento
de computação

```
#include <stdio.h>

int main(){
    printf("Olá, Mundo! \n");
    return 0;
}
```

- Todos os comandos em C devem ser acompanhados de um ponto e vírgula (;) indicando que ali termina aquele comando

PRIMEIRO PROGRAMA EM C++



UFOP



departamento
de computação

```
#include <stdio.h>

int main(){
    printf("Olá, Mundo!\n");
    return 0;
}
```

- O **int** antes da definição **main()** indica que a função **main()** vai retornar um tipo de dado **int**;



PRIMEIRO PROGRAMA EM C++



UFOP



decom

departamento
de computação

```
#include <stdio.h>
```

```
int main(){  
    printf("Olá, Mundo! \n");  
    return 0;  
}
```

- A função **main** é obrigatória em todos os programas!



PRIMEIRO PROGRAMA EM C++



UFOP



departamento
de computação

```
#include <stdio.h>

int main(){
    printf("Olá, Mundo! \n");
    return 0;
}
```

Este é o resultado do programa:



Olá, Mundo!

COMENTÁRIOS



UFOP

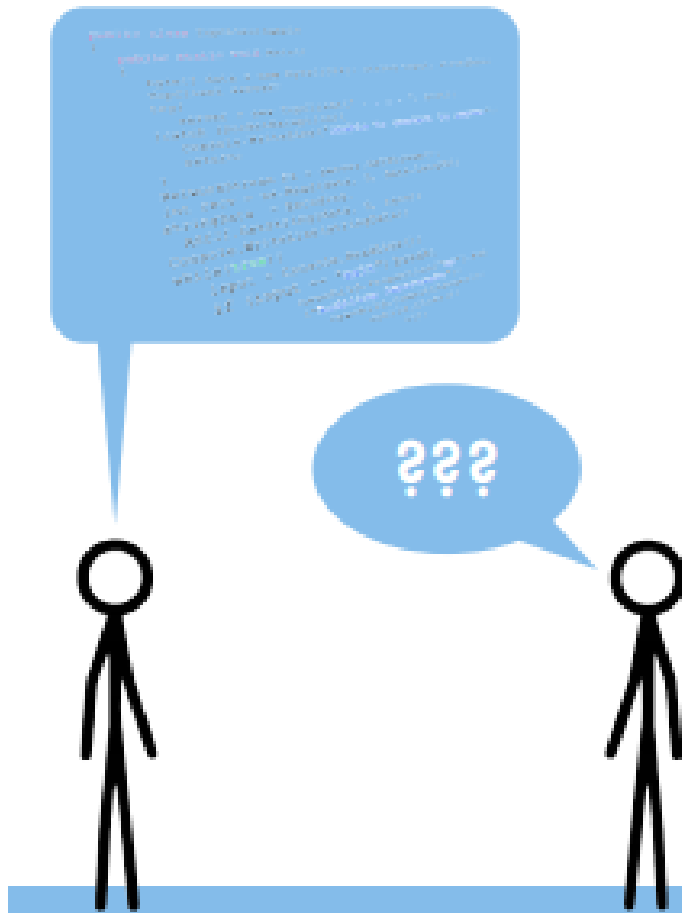


decom

departamento
de computação

Algumas vezes, um código pode se tornar tão complexo nem mesmo um outro programador consiga entender.

Neste caso, seria bom deixar comentários para outros programadores.



COMENTÁRIOS



UFOP



decom

departamento
de computação

- Comentários são parte do código que são ignoradas pelo compilador
- Servem para deixar lembretes ou informações sobre o código para outros programadores
- O comando `//` marca o início de um comentário de linha
- Os comandos `/*` e `*/` são utilizados para iniciar e encerrar um bloco de comentários

COMENTÁRIOS



```
#include <stdio.h>
```

```
int main(){
```

```
    printf("Olá, Mundo! \n"); //Tudo isto é ignorado
```

```
    // printf("Olá, Mundo! \n");
```

```
    printf("Olá! Mais uma vez! \n");
```

```
    /*Aqui se inicia um bloco de comentários.
```

```
    Tudo dentro do bloco é ignorado. */
```

```
    return 0;
```

```
}
```



COMENTÁRIOS



UFOP



decom

departamento
de computação

```
#include <stdio.h>
```

```
int main(){
```

```
    printf("Olá, Mundo! \n"); //Tudo isto é ignorado
```

```
    // printf("Olá, Mundo! \n");
```

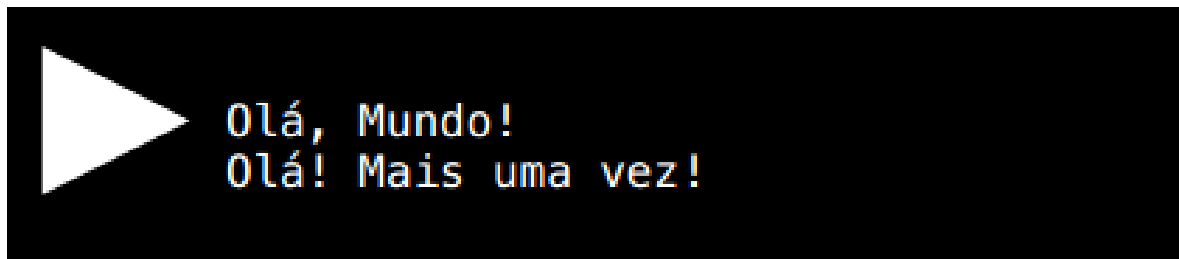
```
    printf("Olá! Mais uma vez! \n");
```

```
    /*Aqui se inicia um bloco de comentários.
```

```
    Tudo dentro do bloco é ignorado. */
```

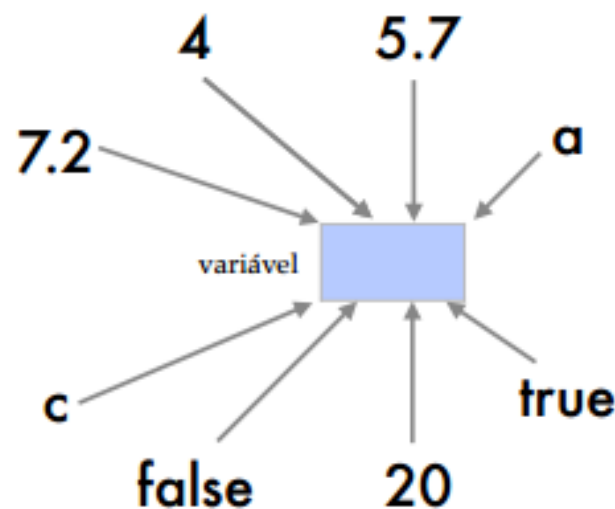
```
    return 0;
```

```
}
```



CRIAÇÃO DE VARIÁVEIS

- Os dados de nosso programa são guardados em variáveis
- É necessário informar qual o tipo de dado de cada variável
- Damos para cada variável um nome identificador que a referencia



CRIAÇÃO DE VARIÁVEIS



UFOP



decom

departamento
de computação

```
#include <stdio.h>
```

```
int main(){  
    int numero; //criando uma variável do tipo int com nome numero  
    char letra; //criando uma variável do tipo char com nome letra  
    double num_real; //criando uma variável do tipo double com nome num_real  
    printf("Este programa criou 3 variaveis! \n");  
    return 0;  
}
```

CRIAÇÃO DE VARIÁVEIS



UFOP

departamento
de computação

numero



```
#include <stdio.h>
```

```
int main(){
```

```
    int numero; //criando uma variável do tipo int com nome numero
```

```
    char letra; //criando uma variável do tipo char com nome letra
```

```
    double num_real; //criando uma variável do tipo double com nome num_real
```

```
    printf("Este programa criou 3 variaveis! \n");
```

```
    return 0;
```

```
}
```



CRIAÇÃO DE VARIÁVEIS



UFOP



decom

departamento
de computação

numero



Repare que a variável numero é escrita sem acento!

```
#include <stdio.h>
```

```
int main(){
```

```
    int numero; //criando uma variável do tipo int com nome numero
```

```
    char letra; //criando uma variável do tipo char com nome letra
```

```
    double num_real; //criando uma variável do tipo double com nome num_real
```

```
    printf("Este programa criou 3 variaveis! \n");
```

```
    return 0;
```

```
}
```





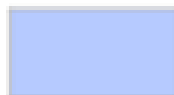
UFOP



departamento
de computação

CRIAÇÃO DE VARIÁVEIS

numero



letra



```
#include <stdio.h>
```

```
int main(){
```

```
    int numero; //criando uma variável do tipo int com nome numero
```

```
    char letra; //criando uma variável do tipo char com nome letra
```

```
    double num_real; //criando uma variável do tipo double com nome num_real
```

```
    printf("Este programa criou 3 variaveis! \n");
```

```
    return 0;
```

```
}
```





UFOP



departamento
de computação

CRIAÇÃO DE VARIÁVEIS

numero  letra  num_real 

```
#include <stdio.h>
```

```
int main(){  
    int numero; //criando uma variável do tipo int com nome numero  
    char letra; //criando uma variável do tipo char com nome letra  
    double num_real; //criando uma variável do tipo double com nome num_real  
    printf("Este programa criou 3 variaveis! \n");  
    return 0;  
}
```



UFOP



decom

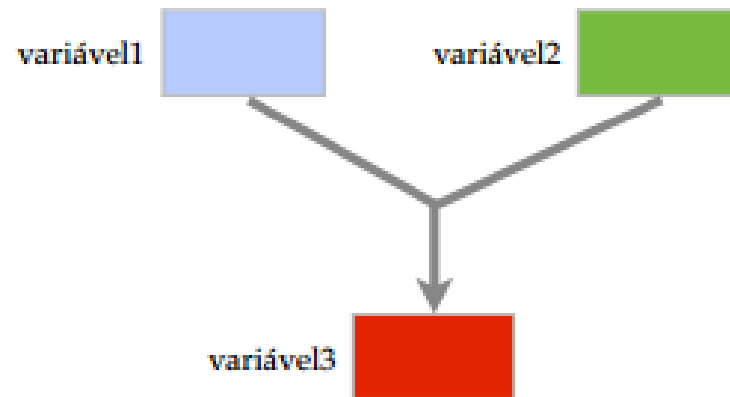
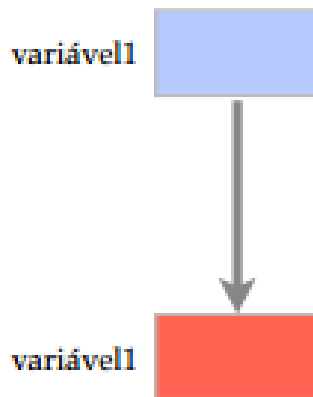
departamento
de computação

IDENTIFICADORES

- Nome que damos a uma variável ou função (estudaremos mais adiante)
- Regras para criar um nome:
 - O primeiro caractere deve ser uma letra ou um sinal de sublinha (_)
 - Exemplos:
 - numero
 - Numero
 - _numero
 - 1numero (errado)
 - @email (errado)
- Há diferenciação entre maiúsculas e minúsculas: numero é diferente de Numero. (Case sensitive)

OPERADORES

- Operadores são muito importantes em programação pois eles alteram o valor de variáveis ou usam variáveis existentes para criar novas variáveis





UFOP



decom

departamento
de computação

OPERADOR DE ATRIBUIÇÃO

- Operador =
- Não confundir com sinal de igualdade
 - O operador = não diz que dois valores são iguais
 - O operador = atribui o valor à direita à variável à esquerda
- Dá um valor a uma variável
- O valor da variável pode ser utilizado posteriormente
- O valor antigo da variável é perdido



UFOP



decom

departamento
de computação

OPERADOR DE ATRIBUIÇÃO

```
#include <stdio.h>
```

```
int main(){
```

```
    int numero; //criando uma variável do tipo int com nome numero
```

```
    numero = 4;
```

```
    printf("A variavel numero do tipo int tem valor %d \n", numero);
```

```
    return 0;
```

```
}
```



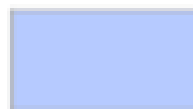
UFOP



departamento
de computação

OPERADOR DE ATRIBUIÇÃO

numero



A variável é criada.

```
#include <stdio.h>
```

```
int main(){
```

```
    int numero; //criando uma variável do tipo int com nome numero
```

```
    numero = 4;
```

```
    printf("A variavel numero do tipo int tem valor %d \n", numero);
```

```
    return 0;
```

```
}
```



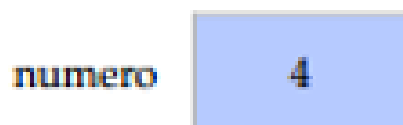


UFOP



departamento
de computação

OPERADOR DE ATRIBUIÇÃO



```
#include <stdio.h>
```

A variável é inicializada.

```
int main(){
```

```
    int numero; //criando uma variável do tipo int com nome numero
```

```
    numero = 4;
```

```
    printf("A variavel numero do tipo int tem valor %d \n", numero);
```

```
    return 0;
```

```
}
```





UFOP



departamento
de computação

OPERADOR DE ATRIBUIÇÃO



```
#include <stdio.h>
```

A variável é inicializada.

```
int main(){
```

```
    int numero; //criando uma variável do tipo int com nome numero
```

```
    numero = 4;
```

```
    printf("A variavel numero do tipo int tem valor %d \n", numero);
```

```
    return 0;
```

```
}
```

O operador = diz que a variável numero receberá o valor 4. Esse operador não compara a variável ao valor 4.



UFOP



departamento
de computação

OPERADOR DE ATRIBUIÇÃO



```
#include <stdio.h>
```

```
int main(){
```

```
    int numero; //criando uma variável do tipo int com nome numero
```

```
    numero = 4;
```

```
    printf("A variavel numero do tipo int tem valor %d \n", numero);
```

```
    return 0;
```

```
}
```

Ao imprimir a mensagem, a palavra **numero** é substituída pelo valor da variável **numero**



A variável numero do tipo int tem valor 4

FLUXO DE ENTRADA



UFOP

departamento
de computação

- O objeto **scanf** é utilizado para que o usuário possa fazer entradas no programa



FLUXO DE ENTRADA



UFOP



decom

departamento
de computação

- Isto é importante para interagir com a pessoa que utiliza o programa
- Assim, ela é complementar à opção de atribuição = para dar valor a variáveis



UFOP

departamento
de computação

FLUXO DE ENTRADA

```
#include <stdio.h>
```

```
int main(){  
    float numero1;  
    float numero2;  
    float soma;  
    printf("Digite o primeiro numero: ");  
    scanf("%f", &numero1);  
    printf("Digite o segundo numero: ");  
    scanf("%f", &numero2);  
    soma = numero1 + numero2;  
    printf("A soma dos numeros e %f \n", soma);  
    return 0;  
}
```





UFOP



departamento
de computação

FLUXO DE ENTRADA

Comandos que criam as variáveis.

```
#include <stdio.h>
```

```
int main(){
```

```
    float numero1;
```

```
    float numero2;
```

```
    float soma;
```

```
    printf("Digite o primeiro numero: ");
```

```
    scanf("%f", &numero1);
```

```
    printf("Digite o segundo numero: ");
```

```
    scanf("%f", &numero2);
```

```
    soma = numero1 + numero2;
```

```
    printf("A soma dos numeros e %f \n", soma);
```

```
    return 0;
```

```
}
```

numero1





UFOP



departamento
de computação

FLUXO DE ENTRADA

Comandos que criam as variáveis.

```
#include <stdio.h>
```

```
int main(){
```

```
    float numero1;
```

```
    float numero2;
```

```
    float soma;
```

```
    printf("Digite o primeiro numero: ");
```

```
    scanf("%f", &numero1);
```

```
    printf("Digite o segundo numero: ");
```

```
    scanf("%f", &numero2);
```

```
    soma = numero1 + numero2;
```

```
    printf("A soma dos numeros e %f \n", soma);
```

```
    return 0;
```

```
}
```

numero1

numero2



UFOP



departamento
de computação

FLUXO DE ENTRADA

Comandos que criam as variáveis.

```
#include <stdio.h>
```

```
int main(){
```

```
    float numero1;
```

```
    float numero2;
```

```
    float soma;
```

```
    printf("Digite o primeiro numero: ");
```

```
    scanf("%f", &numero1);
```

```
    printf("Digite o segundo numero: ");
```

```
    scanf("%f", &numero2);
```

```
    soma = numero1 + numero2;
```

```
    printf("A soma dos numeros e %f \n", soma);
```

```
    return 0;
```

```
}
```





UFOP



departamento
de computação

FLUXO DE ENTRADA

Imprime uma mensagem pedindo ao usuário para executar uma ação.

```
#include <stdio.h>
```

```
int main(){  
    float numero1;  
    float numero2;  
    float soma;  
    printf("Digite o primeiro número: ");  
    scanf("%f", &numero1);  
    printf("Digite o segundo número: ");  
    scanf("%f", &numero2);  
    soma = numero1 + numero2;  
    printf("A soma dos números é %f \n", soma);  
    return 0;  
}
```

numero1

numero2

soma





UFOP



departamento
de computação

FLUXO DE ENTRADA

Repare que não um comando `\n`
para passar para a próxima linha.

```
#include <stdio.h>
```

```
int main(){  
    float numero1;  
    float numero2;  
    float soma;  
    printf("Digite o primeiro número: ");  
    scanf("%f", &numero1);  
    printf("Digite o segundo número: ");  
    scanf("%f", &numero2);  
    soma = numero1 + numero2;  
    printf("A soma dos números é %f \n", soma);  
    return 0;  
}
```

numero1	<input type="text"/>
numero2	<input type="text"/>
soma	<input type="text"/>





UFOP



departamento
de computação

FLUXO DE ENTRADA

Scanf é usado para obter um valor do teclado

```
#include <stdio.h>
```

```
int main(){  
    float numero1;  
    float numero2;  
    float soma;  
    printf("Digite o primeiro numero: ");  
    scanf("%f", &numero1);  
    printf("Digite o segundo numero: ");  
    scanf("%f", &numero2);  
    soma = numero1 + numero2;  
    printf("A soma dos numeros e %f \n", soma);  
    return 0;  
}
```

numero1	<input type="text"/>
numero2	<input type="text"/>
soma	<input type="text"/>





UFOP



departamento
de computação

FLUXO DE ENTRADA

O programa fica parado até que o usuário digite um valor e aperte a tecla Enter

```
#include <stdio.h>
```

```
int main(){  
    float numero1;  
    float numero2;  
    float soma;  
    printf("Digite o primeiro numero: ");  
    scanf("%f", &numero1);  
    printf("Digite o segundo numero: ");  
    scanf("%f", &numero2);  
    soma = numero1 + numero2;  
    printf("A soma dos numeros e %f \n", soma);  
    return 0;  
}
```

numero1	<input type="text"/>
numero2	<input type="text"/>
soma	<input type="text"/>





UFOP



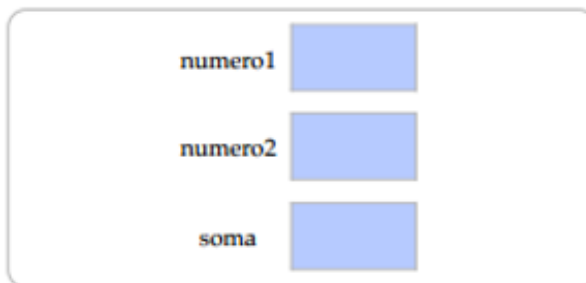
departamento
de computação

FLUXO DE ENTRADA

Suponha que o valor 3.14 tenha sido digitado

```
#include <stdio.h>
```

```
int main(){  
    float numero1;  
    float numero2;  
    float soma;  
    printf("Digite o primeiro numero: ");  
    scanf("%f", &numero1);  
    printf("Digite o segundo numero: ");  
    scanf("%f", &numero2);  
    soma = numero1 + numero2;  
    printf("A soma dos numeros e %f \n", soma);  
    return 0;  
}
```





UFOP



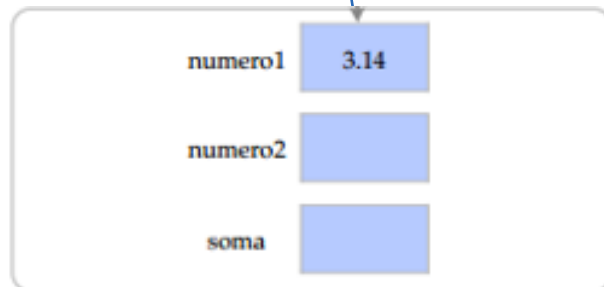
departamento
de computação

FLUXO DE ENTRADA

3.14 é guardado na variável e vamos para o próximo passo.

```
#include <stdio.h>
```

```
int main(){  
    float numero1;  
    float numero2;  
    float soma;  
    printf("Digite o primeiro numero: ");  
    scanf("%f", &numero1);  
    printf("Digite o segundo numero: ");  
    scanf("%f", &numero2);  
    soma = numero1 + numero2;  
    printf("A soma dos numeros e %f \n", soma);  
    return 0;  
}
```





UFOP



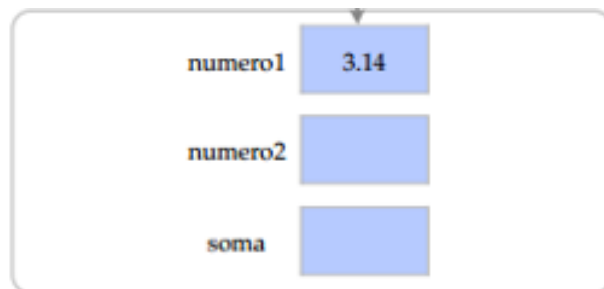
departamento
de computação

FLUXO DE ENTRADA

O programa espera a entrada de dados novamente.

```
#include <stdio.h>
```

```
int main(){  
    float numero1;  
    float numero2;  
    float soma;  
    printf("Digite o primeiro numero: ");  
    scanf("%f", &numero1);  
    printf("Digite o segundo numero: ");  
    scanf("%f", &numero2);  
    soma = numero1 + numero2;  
    printf("A soma dos numeros e %f \n", soma);  
    return 0;  
}
```





UFOP



departamento
de computação

FLUXO DE ENTRADA

Atribuiremos a variável soma o valor de `numero1 + numero2`.

```
#include <stdio.h>
```

```
int main(){  
    float numero1;  
    float numero2;  
    float soma;  
    printf("Digite o primeiro numero: ");  
    scanf("%f", &numero1);  
    printf("Digite o segundo numero: ");  
    scanf("%f", &numero2);  
    soma = numero1 + numero2;  
    printf("A soma dos numeros e %f \n", soma);  
    return 0;  
}
```

numero1 3.14

numero2 2.7

soma



Digite o primeiro número: 3.14
Digite o segundo número: 2.7



UFOP



departamento
de computação

FLUXO DE ENTRADA

Efetua-se a soma.

```
#include <stdio.h>
```

```
int main(){  
    float numero1;  
    float numero2;  
    float soma;  
    printf("Digite o primeiro numero: ");  
    scanf("%f", &numero1);  
    printf("Digite o segundo numero: ");  
    scanf("%f", &numero2);  
    soma = numero1 + numero2;  
    printf("A soma dos numeros e %f \n", soma);  
    return 0;  
}
```

numero1 3.14

numero2 2.7

soma 5.84



Digite o primeiro número: 3.14
Digite o segundo número: 2.7



UFOP



departamento
de computação

FLUXO DE ENTRADA

É impresso na tela o resultado da soma.

```
#include <stdio.h>
```

```
int main(){  
    float numero1;  
    float numero2;  
    float soma;  
    printf("Digite o primeiro numero: ");  
    scanf("%f", &numero1);  
    printf("Digite o segundo numero: ");  
    scanf("%f", &numero2);  
    soma = numero1 + numero2;  
    printf("A soma dos numeros e %f \n", soma)  
    return 0;  
}
```

numero1 3.14

numero2 2.7

soma 5.84



```
Digite o primeiro número: 3.14  
Digite o segundo número: 2.7  
A soma dos números é 5.84
```



UFOP



departamento
de computação

FLUXO DE ENTRADA

Fim do programa.

```
#include <stdio.h>
```

```
int main(){  
    float numero1;  
    float numero2;  
    float soma;  
    printf("Digite o primeiro numero: ");  
    scanf("%f", &numero1);  
    printf("Digite o segundo numero: ");  
    scanf("%f", &numero2);  
    soma = numero1 + numero2;  
    printf("A soma dos numeros e %f \n", soma)  
    return 0;  
}
```

numero1 3.14

numero2 2.7

soma 5.84



Digite o primeiro número: 3.14
Digite o segundo número: 2.7
A soma dos números é 5.84



UFOP



decom

departamento
de computação

BIBLIOGRAFIA

- CORMEN, Thomas; RIVEST, Ronald, STEIN, Clifford, LEISERSON, Charles. Algoritmos. Rio de Janeiro: Elsevier, 2002.
- DEITEL, Paul; DEITEL, Harvey. C++ como programar. 5. ed. São Paulo: Pearson, 2006.
- MELO, Ana Cristina Vieira de; SILVA, Flávio Soares Corrêa da. Princípios de linguagens de programação. São Paulo: Edgard Blücher, 2003.
- ASCENCIO, A. F. G. & CAMPOS, E. A. V. Fundamentos da programação de computadores. 2. ed. São Paulo: Pearson Education, 2007.
- MEDINA, Marcelo, FERTIG, Cristina. Algoritmos e programação: teoria e prática. Novatec. 2005.
- MIZRAHI, V. V.. Treinamento em linguagem C: módulo 1. São Paulo: Makron Books, 2008.
- PUGA, S. & RISSETTI, G. Lógica de programação e estruturas de dados com aplicações em java. São Paulo: Prentice Hall, 2004.
- ZIVIANI, Nívio. Projeto de Algoritmos com Implementação em Pascal e C. Cengage Learning. 2010.