

# Programação em C

Prof. Eduardo Magalhães

# Linguagem C

- Histórico:
- ✓ Surgiu na década de 70 criado por Dennis Ritchie, implementou-o pela primeira vez usando um DEC PDP-11;
- ✓ Rodava no SO Unix;
- ✓ Derivou de uma outra linguagem “ B” criada por Ken Thompson.

# Linguagem C

- Características:
  - ✓ Linguagem de programação de baixo e alto nível;
  - ✓ Usada para criação de diversos softwares como: processadores de texto, planilhas eletrônicas, sistemas operacionais, programas de automação industrial, gerenciadores de banco de dados, navegadores e muitos outros.

# Linguagem C

- Plataformas ou SOs:
  1. Windows:
    - Eclipse (versão para C/C++) e o compilador de C é MinGW;
    - Netbeans versão IDE com suporte a C/C++;
    - DEV C++ (versão para C/C++) e o compilador de C é MinGW.
    - Falcon (versão para C/C++) e o compilador de C é MinGW.

# Linguagem C

- Plataformas ou SOs:

- 2. Linux:

- Eclipse (versão para C/C++) e o compilador de C é GNU C;

- Netbeans com compiladores Sun Studio, gdb e make;

- DEV C++ (versão para C/C++) e o compilador de C é GCC (GNU Compiler Collection);

- Falcon (versão para C/C++) e o compilador de C é MinGW.

# Linguagem C

- A linguagem C é “***case sensitivo***”;

Significa dizer que letras *maiúsculas e minúsculas* fazem diferença.

Se você declarar uma variável com o nome ***soma*** ela será diferente de **Soma**, **SOMA**, **SoMa** ou **sOmA**

# Meu primeiro programa

```
/* Primeiro programa em C */
```

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

```
#include<stdlib.h>
```

```
int main ()
```

```
{
```

```
    printf ("Ola! mundo!\n");
```

```
    return(o);
```

```
}
```

# Entendo o código

- **Linha 1 – `//Primeiro programa em C`:** Comentário de uma linha (a partir do `//` até o fim da linha);
- **Linhas 2 e 3 – `#include<stdlib.h>` e `include<stdio.h>`:** Inclusão de bibliotecas `stdlib` e `stdio` para operações de entrada e saída;
- **Linha 5 – `int main(void) {`:** Declaração da função `main` e abertura do programa;
- **Linha 6 – `printf(“Ola mundo”);`:** Função `printf`, responsável por escrever a mensagem “Olá mundo” na tela;
- **Linha 7 – `return 0;`:** Retorno da função principal (0);
- **Linha 8 – `}`:** Fechamento do corpo da função principal (finalização do programa).



# Geração de executável

## ATENÇÃO!

- Para que possamos executar nosso programa, primeiro é necessário compilá-lo, ou seja, gerar o executável correspondente ao código fonte digitado.

# Formatação

- **Código Tipo de variável:**
  - %d Inteiro (int)
  - %f Real (float)
  - %c Caractere (char)
  - %s Cadeia de caracteres (char[])

# Inserção de caracteres

- É possível, ainda, passar parâmetros capazes de imprimir caracteres especiais, necessários, especialmente, para formatarmos a saída do nosso programa.
- Por exemplo: `\n` (quebra de linha), `\t` (tabulação), `\\` (escreve a barra), entre outros.

# Operações de I/O

- **#include <stdio.h>** diz ao compilador que ele deve incluir o arquivo-cabeçalho **stdio.h**;
- Neste arquivo existem declarações de funções úteis para entrada e saída de dados (std = standard, padrão em inglês; io = Input/Output, entrada e saída ==> stdio = Entrada e saída padronizadas).

# Comentários

- Quando fazemos um programa, uma boa idéia é usar comentários que ajudem a elucidar o funcionamento do mesmo;

Exemplo: :

```
/* Meu Primeiro Programa */
```

# Função principal

- A linha **int main()** indica que estamos definindo uma função de nome **main**.
- Todos os programas em C têm que ter uma função **main**, pois é esta função que será chamada quando o programa for executado.

# Função principal

- O conteúdo da função é delimitado por chaves `{ }`. O código que estiver dentro das chaves será executado seqüencialmente quando a função for chamada.
- A palavra `int` indica que esta função retorna um inteiro. O que significa este retorno será visto posteriormente, quando estudarmos um pouco mais detalhadamente as funções do C.

# Função Return

- A última linha do programa, **return(o);** , indica o número inteiro que está sendo retornado pela função, no caso o número 0 (zero).



# Relação Printf e Stdio

- A única coisa que o programa *realmente* faz é chamar a função **printf()**, passando a string (uma string é uma seqüência de caracteres, como veremos brevemente);
- É por causa do uso da função **printf()** pelo programa que devemos incluir o arquivo-cabeçalho **stdio.h**

# O uso do \n

- o `\n` é a constante barra invertida de "new line" e ele é interpretado como um comando de mudança de linha.

# Programa Conversão Dias em Anos

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

```
int main () {
```

```
    int Dias; /* Declaracao de Variaveis */
```

```
    float Anos;
```

```
    printf ("Entre com o número de dias: "); /* Entrada de  
    Dados */
```

```
    scanf ("%d",&Dias);
```

```
    Anos=Dias/365.25; /* Conversao Dias->Anos */
```

```
    printf      ("\n\n%d      dias      equivalem      a      %f  
    anos.\n",Dias,Anos); return(o);
```

```
}
```

# O que faz este programa?

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

```
int main ()
```

```
{
```

```
    int x;
```

```
    scanf("%d",&x);
```

```
    printf("%d",x);
```

```
    return(o);
```

```
}
```

## EXERCICIOS PROPOSTOS

- 1. Faça um programa que escreva seu nome.
- 2. Faça um programa que escreva suas notas em linhas diferentes com exatamente uma casa decimal. Use a formatação `%.1f` para substituir pelos valores e, a partir disso, explique como esse marcador funciona.
- 3. Faça um programa que escreva seu nome e sobrenomes, cada um em uma linha.

## EXERCÍCIOS PROPOSTOS

- 5. Faça um programa que calcule o resultado da expressão:  $(537 - 285) * 10 + (3 * (72 - 17))$ .
- 6. Faça um programa que receba um número inteiro e calcule o resto da divisão por 2.
- 7. Faça um programa que calcule o IMC de uma pessoa.