

Conceitos a serem utilizados:**Estrutura de um programa em C - esqueleto**

```

/* comentário inicial, descrevendo objetivos do
programa */
#include <stdlib.h> // para usar system
#include <stdio.h> // para entrada e saída de dados
// outras bibliotecas necessárias
// outras declarações

int main() // função principal e obrigatório
{
    //declarações e comandos//

    system("pause"); /* ou system("pause >> null");
                     - para eliminar mensagem */
    return 0; // para encerramento normal do programa
}

```

Declarações

Programa principal

Slide 29

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

nome de variável: iniciar por uma letra, composto de até 32 caracteres (letras, dígitos, sublinhado) . Usar minúsculas!

declarações de variáveis:

tipo <nomevar₁>;

tipo <nomevar₂>, ... , <nomevar_n>;

tipo: int, float, double, char.

Bibliotecas: permitem inclusão de funções pré-definidas da linguagem.

#include <stdio.h>	Funções de entrada e saída
#include <stdlib.h>	Funções padrão
#include <math.h>	Funções matemáticas
#include <system.h>	Funções do sistema
#include <string.h>	Funções de texto

Funções e operador a serem utilizados: scanf - atribuição - printf**Iniciando e Encerrando atividades práticas no Dev-C++**

(para baixar: <http://www.bloodshed.net/devcpp.html>)

- 1 - Ligar o computador;
- 2 - Quando aparecer a janela com usuário ..., digitar <enter> em lugar da senha;
- ⇒ 3 - Na área adequada do harddisk ou outro dispositivo pessoal, **crie um diretório pessoal** para armazenar os trabalhos de aula;
- 4 - Inicie o Dev-C++ pelo ícone ou pelo menu Iniciar;
- 5 - Quando **terminar** de utilizar o C lembre-se de **SEMPRE**:
 1. **fechar todos os programas** do C
 2. sair do Dev-C++ - opção **Arquivo/Sair**
 3. no canto inferior à esquerda: **Iniciar** ⇒ **desligar o computador**

Programando em Dev-C++

a) Através de **Arquivo/Novo/Arquivo Fonte**, abra um novo fonte a ser criado.

b) Os principais itens do primeiro menu são:

- **Novo/Arquivo Fonte**: cria um novo arquivo fonte
- **Abrir Projeto ou Arquivo**: abre um arquivo fonte já existente
- **Salvar e Salvar como...**: salva o fonte com o nome já definido (salvar) ou salva com o nome aqui definido (salvar como)
- **Imprimir**: imprime o código fonte
- **Configurar Impressão**: mostra as configurações da impressora
- **Sair**: sai do Dev-C++ .
- Os **Reabrir** identifica os últimos arquivos fonte utilizados.

c) **Editar** o programa Dev-C++ abaixo, empregando a opção **Novo/Arquivo Fonte**:

```
/* Primeiro programa - impressão de mensagem*/
#include <stdio.h> /* funções padrão de entrada e saída*/
int main()        /* função obrigatória, chamada na
                  execução do programa*/
{
    printf ("Alo mundo!");    // coloca conteúdo na tela
    return 0;                // término normal da função main
}
```

d) **Salvar** o programa com o nome **exemplo1a mensagem** - para tanto, selecione o menu **Arquivo /Salvar como**.

e) **Observe agora as instruções abaixo**:

- Compile o programa com o comando **Executar/Compilar** ou com a tecla **Ctrl-F9**.
- Se houver algum erro de sintaxe, aparece uma ou mais mensagens no rodapé da janela. Neste caso, corrija o programa e repita.
- Se não houver erros, execute o programa com o comando **Executar/Executar** ou com a tecla **Ctrl-F10**

g) Observe que o programa encerrou antes da tela poder ser visualizada. Retorne ao programa fonte, salve novamente o programa alterando o nome para **exemplo1b mensagem** e inclua 2 novas linhas, de forma que este fique na forma abaixo:

```
/* Primeiro programa - impressão de mensagem*/
#include <stdio.h>
#include <stdlib.h>
int main()
{
    printf ("Alo mundo!");
    system("pause");
    return 0;
}
```

f) Para encerrar, altere o programa acima, para que a mensagem “Pressione qualquer tecla....” apareça na nova linha, separada de “Alo mundo!”.

DICAS IMPORTANTES:

Termine todas as linhas com **;** .

Sempre salve o programa antes de compilar.

Sempre compile o programa antes de executar.

Quando ocorrer um **erro de compilação**, dê um duplo clique sobre a mensagem de erro para destacar o comando errado no programa.

Verifique também **a linha anterior**, que pode ser a responsável pelo erro, especialmente se faltar o **;**.

Use comentários, iniciados por **//** ou delimitado por **/* */** .

Atribuições entre tipos diferentes: se um valor for atribuído a uma variável de tipo diferente, então deve ocorrer uma conversão durante a execução deste comando. Entretanto, algumas vezes, a conversão resulta na perda de informação. Por exemplo:

No **scanf**, omitir o operador de endereço **&** antes de um identificador causa o **CANCELAMENTO** do programa.

Declarar uma variável de um tipo (p. ex., **int**) e no **scanf** usar um formato que não corresponde (p. ex., **%f**) faz com que **o programa seja CANCELADO ou os valores sejam alterados!**

Problema 1: Faça um programa contendo a declaração de 4 variáveis, cada uma com um tipo diferente (int, float, double e char). Armazene valores diferentes para cada uma delas, adequado ao conteúdo esperado, através de comandos de leitura ou atribuição. Imprima e analise os resultados.

Acrescente as linhas abaixo, observando as diferentes forma como os conteúdos reais podem ser visualizados através da formatação inserida entre os símbolos % e f.

```
printf ("O valor %f \te h o exemplo 1\n", 157.8926);  
printf ("O valor %9.2f \te h o exemplo 2\n", 157.8926);  
printf ("O valor %+8.2f \te h o exemplo 3\n", 157.8926);  
printf ("O valor %7.5f \te h o exemplo 4\n", 157.8926);  
printf ("O valor %e \te h o exemplo 5\n", 157.8926);  
printf ("O valor %.3E \te h o exemplo 6\n", 157.8926);  
printf ("O valor %g \te h o exemplo 7\n\n", 157.8926);
```

Problema 2: Faz o programa em C que, dado o preço de um produto em reais, converta este valor para o equivalente em dólares. O programa deverá ler do teclado o preço do produto e a taxa de conversão para o dólar. A tela deve assemelhar-se a:

