

Aula 9

Prof. Daniel Cavalcanti Jeronymo

Fundamentos de Programação

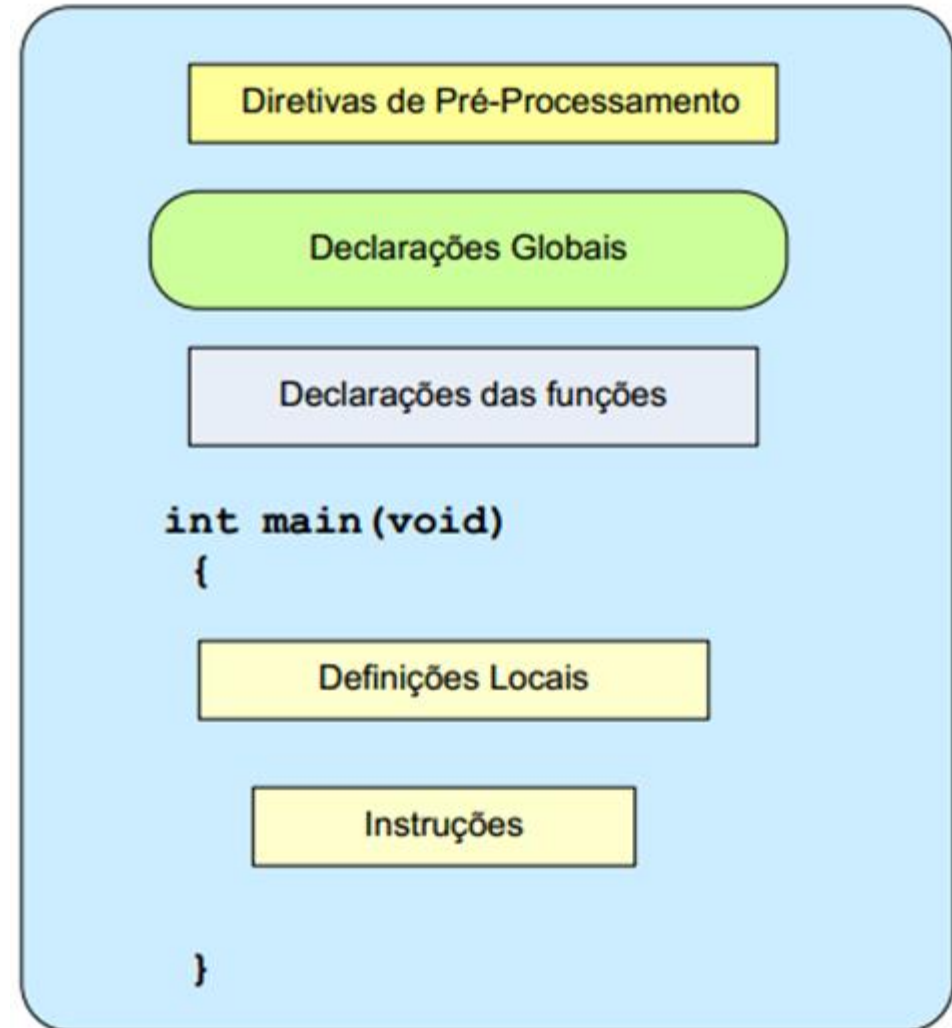
CP41F

Blocos e escopo. Declarações.
Estruturas de seleção (if, else, else if,
switch). goto.

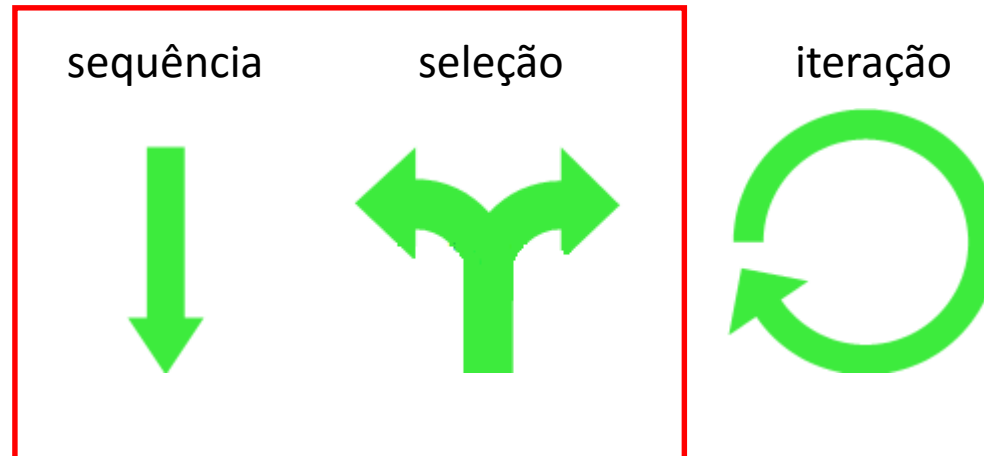
Universidade Tecnológica Federal do Paraná (UTFPR)
Engenharia de Computação – 1º Período

- Revisão
 - Blocos e escopo, declaração e seleção
- Transferência de controle (goto)
- Estrutura de seleção (if, else if, else)
- Estrutura de seleção (switch)

- Blocos
 - Agrupamento de declarações
 - Definição de escopo
 - Delimitado por colchetes { }



- Programação Estruturada
 - Combinação de estruturas de controle



- Transferência de controle (goto)
 - Realiza saltos entre pontos identificados (labels) do código

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

```
int main()
```

```
{
```

```
    goto label2;
```

```
label1:
```

```
    printf("1\n");
```

```
    goto label4;
```

```
label2:
```

```
    printf("2\n");
```

```
label3:
```

```
    printf("3\n");
```

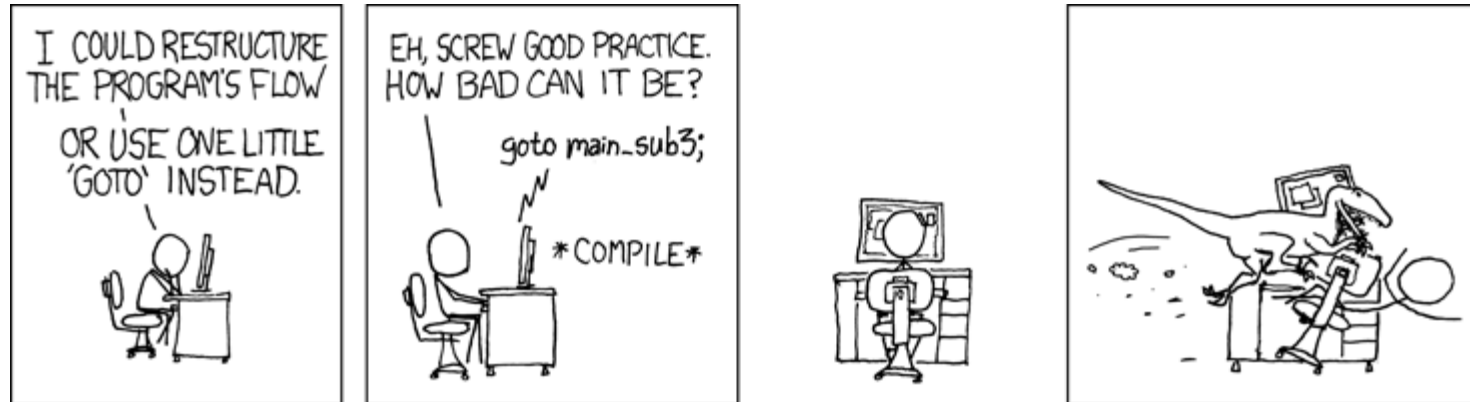
```
    goto label1;
```

```
label4:
```

```
    return 0;
```

```
}
```

- Transferência de controle (goto)
 - Não usar! Leva a código espaguete. Alternativa: programação estruturada.



- Leituras recomendadas:

Dijkstra, Go To Statement Considered Harmful, 1968.

Knuth, Structured Programming with go to Statements, 1974.

If - else if - else

- Se a condição é VERDADEIRA, executa o bloco imediatamente a seguir. Se a condição é FALSA, o bloco é ignorado.

```
if(condicao)  
  BLOCO
```

```
if(a == 5)  
  printf("A é igual a 5!\n");
```

If - else if - else

- Erro comum:

```
if(a == 5)
    printf("A é igual");
    printf(" a 5!\n");
```

Errado!

```
if(a == 5)
{
    printf("A é igual");
    printf(" a 5!\n");
}
```

Correto!

If - else if - else

- Else if: Verificado quando outras condições “if” e “else if” anteriores falharam

```
if(condicao1)
```

```
    BLOCO1
```

```
else if(condicao2)
```

```
    BLOCO2
```

```
else if(condicao3)
```

```
    BLOCO3
```

If - else if - else

- Else: Executa quando todas as outras alternativas falharam
- Equivale à negação lógica das hipóteses anteriores

```
if(condicao1)
```

```
    BLOCO1
```

```
else if(condicao2)
```

```
    BLOCO2
```

```
else if(condicao3)
```

```
    BLOCO3
```

```
else
```

```
    BLOCO4
```

Switch

- Pode ser utilizado para substituir grandes cadeias de if-elseif-else
- Cada caso é um label, a estrutura equivale à uma *“jump table”*

```
switch(variavel)
{
    case 0:
    {
        printf("A variavel e 1!\n");
    }break:
    case 3:
    {
        printf("A variavel e 1!\n");
    }break:
    default:
        printf("A variavel nao e nenhuma das opcoes!\n");
}
```

Switch

- Permite execução encadeada (*fall through*)
- *break* é utilizado para encerrar o caso
- Casos não precisam estar ordenados

```
switch(variavel)
{
  case 0:
  {
    printf("Caso 0!\n");
  }
  case 3:
  {
    printf("Caso 3!\n");
  }
  default:
    printf("Caso padrao!\n");
}
```

- *fall through* ocorre pois o switch **não seleciona um bloco de código** mas sim **salta para a linha de código e passa a executá-la**
- Aplicação interessante: Dispositivo de Duff (Duff's device)