

# PROGRAMAÇÃO I

---

**Prof. Luiz Albano**

@ luiz.albano@ifsp.edu.br



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
SÃO PAULO  
Campus Bragança Paulista

# Agenda

Tema: Estrutura de Seleção

- SWITCH
- BREAK



# Introdução

A linguagem C tem uma estrutura de múltipla seleção, chamada **switch**. Esta estrutura testa sucessivamente o valor de uma expressão contra uma lista de constantes (inteiros, números decimais ou caractere). Quando o valor coincide, os comandos associados àquela constante são executados.

A instrução **switch** adapta-se particularmente à tomada de decisões em que o número de possibilidades é elevado (em geral maior que 2, se não usa-se a estrutura *if-else*), de forma a reduzir a complexidade de *if-else* consecutivos e encadeados.



# Switch

## Sintaxe:

```
switch ( <expressão> ) {  
    case <constante1>:  
        <sequencia_de_instruções>;  
        break;  
    case <constante2>:  
        <sequencia_de_instruções>;  
        break;  
    ...  
    default:  
        <sequencia_de_instruções>;  
}
```



# Switch

Na sintaxe anterior o valor **<expressão>** (qualquer expressão cujo resultado seja um valor dos tipos **char**, **int** ou **float**) é avaliada e, em seguida, a estrutura compara o valor da expressão com o valor de cada **<constante<sub>x</sub>>** que segue declarada em cada um dos **case**.

Quando uma **expressão** coincide com o valor de um **case**, a sequência de comandos associada àquele case será executada até o que o comando **break** ou o fim do comando **switch** seja alcançado.

O bloco **default** é opcional e, se não estiver presente, nenhuma ação será realizada se todos os testes falharem.



# Switch

## NOTA:

Em cada case do switch só uma única constante do tipo char, int ou float pode estar presente para avaliação.



# Switch

## Funcionamento do switch (De novo!)

- Se o valor da **expressão** for igual a alguma das constantes que seguem os vários **case**, então são executadas as instruções que seguem o **case** correspondente;
- Se o valor da **expressão** não for igual a nenhuma das constantes apresentadas pelos **case**, então são executadas as instruções que seguem o **default**;
- O bloco **default** é opcional. No caso de o valor da expressão não ser igual a nenhum dos **case**, nada é executado, terminando o **switch**. O programa continua na instrução seguinte ao **switch**.



# Switch

## NOTA:

Na instrução switch, quando a expressão é igual a uma das constantes presentes em um dos case a instrução ou instruções associadas a esse case são executadas, **bem como todas as instruções de todos os case que se encontram a seguir ao case de entrada (incluindo o default se estiver declarado).**







# Comando break;

Como apresentado no slide anterior ao entrar em um **case** a sequencia de instruções do **case** de entrada e de todos os cases declarados na sequencia são executados.

Para que apenas as instruções do **case** selecionado (case de entrada) seja executadas devemos interromper a execução da estrutura **switch**. Para isto utilizamos a instrução: **break**;

A instrução **break** permite parar a execução dentro de um **switch**, continuando o programa na instrução seguinte ao bloco **switch**.

# Exemplo

```
#include <stdio.h>
main() {
    int num1, num2, res=0;
    char op;
    printf("Escreva uma expressao: ");
    scanf("%d %c %d", &num1, &op, &num2);
    switch( op ) {
        case '+': res = num1 + num2; break;
        case '-': res = num1 - num2; break;
        case 'x':
        case 'X':
        case '*': res = num1 * num2; break;
        case ':':
        case '\\':
        case '/': res = num1 / num2; break;
    }
    printf("%d %c %d = %d\\n", num1, op, num2, res);
}
```

# Dúvidas?