

**Engenharia da Computação**

[www.eComp.Poli.br](http://www.eComp.Poli.br)

# Conceitos Fundamentais de Programação Imperativa

Disciplina: **DCExt Programação Imperativa**

Prof. Hemir Santiago

[hcs2@poli.br](mailto:hcs2@poli.br)

Material cedido pelo Prof. Joabe Jesus

# Objetivos

- Discutir os conceitos fundamentais de LPI:
  - Saída de Dados (manipular Unidade de Saída)
  - Instruções Básicas (processamento de dados)
  - Entrada de Dados (manipular Unidade de Entrada)

# A Função `printf`

- Utilizada para enviar sequências de caracteres para a **saída padrão** de um programa em C
- Sintaxe:

```
printf ("string de controle", lista de argumentos);
```



O que desejamos  
exibir na

Valores para  
completar **lacunas** da

**Unidade de  
Saída Padrão**

"string de controle"

C:\>\_

Console



# Exemplos

```
printf("Uma linha");
```

```
printf("Uma linha\nDuas linhas");
```

```
printf("Os numeros sao: %d e %d\n", 7, 8);
```

```
printf("%s estah a %d Km de Recife", "Caruaru", 120);
```

```
printf("%d%% de %d = %f\n", p, x, x*(p/100.0));
```

# A Função `printf`

- Caracteres especiais:

<code>\n</code>	avanço de linha
<code>\t</code>	tabulação (tab)
<code>\b</code>	retrocesso (backspace)
<code>\"</code>	aspas duplas
<code>\\</code>	barra

# A Função `printf`

- Códigos de impressão formatada:

<code>%c</code>	caractere
<code>%d</code>	inteiro
<code>%e</code>	notação científica
<code>%f</code>	ponto-flutuante
<code>%o</code>	octal
<code>%x</code>	hexadecimal
<code>%u</code>	inteiro sem sinal
<code>%s</code>	string
<code>%%</code>	o caractere <code>'%'</code>

# A Função `printf`

- O tamanho de campos de impressão é indicado logo após o '%' e antes do tipo do campo:

```
printf("Os alunos são %4d\n", 44);  
printf("Os alunos são %04d\n", 44);  
printf("Parcela de R$ %.2f/mes\n", 1234.5632);  
printf("Parcela de R$ %15.3f/mes\n", 1234.5632);
```

# Exercício

- Escreva um programa que imprima na tela:

um

dois

três

quatro

```
printf("um\n\tdois\n\t\ttres\n\t\t\tquatro");
```



# A função `putchar()`

- A função *putchar()* imprime um caractere na tela
- Ao invés de usar:

`printf("%c", c)`

Usa-se:

`putchar(c)`

# Tipos de Dados

- Numéricos **Inteiros**

---

Tipo	Tamanho	Valores
char	8 bits	-128 a +127
short	16 bits	-32.768 a +32.767
int	32 bits	-2.147.483.648 a + 2.147.483.647
long	64 bits	-9.223.372.036.854.775.808 a +9.223.372.036.854.775.807

– Esses tipos permitem o uso de expressões com os operadores: +, -, \*,  
/, ^, %

# Números com e sem **SINAL**

- C permite que o programador defina se uma variável de tipo numérico deve ou não reservar o bit de sinal (números negativos)
- **Sintaxe (Notação):**
  - signed** <tipo>
  - unsigned** <tipo>
  - Se nenhum modificador for indicado, o compilador C reservará o bit de sinal (**signed**)
- **Exemplo:** signed char

# Tipos de Dados

- Numéricos de **Ponto Flutuante**

Tipo	Tamanho	Valores
float	32 bits	$\pm 3.40282347\text{E}+38$ $\pm 1.40239846\text{E}-45$
double	64 bits	$\pm 1.79769313486231570\text{E}+308$ $\pm 4.94065645841246544\text{E}-324$

– Permitem usar valores como: 3.14159; 2.0; -1.0,...

– Esses tipos permitem o uso de expressões com os operadores: +, -, \*,  
/, ^, %

# A Função scanf

- Semelhante à função printf(), exceto que é utilizada para leitura de dados

- Sintaxe:

```
scanf("string de controle", endereços);
```



O que desejamos  
**obter** da

**Unidade de  
Entrada Padrão**

C:\>\_

Console

Valores dos endereços de  
variáveis que serão  
alteradas com os valores de  
entrada **de acordo com** a  
"string de controle"

# Exemplo

```
int idade = 10;  
printf("Digite sua idade em anos: ");  
scanf("%d", &idade);
```

 endereço da variável

# Endereços de Variáveis

- Um endereço de memória é visto como um número inteiro sem sinal
- O código para exibição de um endereço é %u

- **Exemplo:**

```
int num = 2;
```

```
printf("Valor=%d, endereco=%u", num, &num);
```

# Outro Exemplo

```
int dia, mes, ano;
```

```
printf("Entre com a data (xx/xx/xxxx): ");
```

```
scanf("%d/%d/%d", &dia, &mes, &ano);
```

```
printf("A data foi: %d/%d/%d\n", dia, mes, ano);
```



# Funções `getche` e `getch`

- Em algumas situações, a função `scanf` não se adapta perfeitamente pois é preciso pressionar <enter> depois da entrada
- As funções `getche` e `getch` efetuam a leitura de um caractere e continuam a execução do programa
  - Biblioteca *conio.h*
  - `getche()` apresenta o caractere lido na tela, enquanto que `getch()` não apresenta

```
char c1 = getch();  
char c2 = getche();
```

**OBS:** Pode ser útil no final do programa. Evita fechar janela do console ao terminar execução.

# Exercício

- Escreva um programa que peça para o usuário digitar um caractere na tela, imprima o caractere digitado na mesma linha e, por fim, imprima em linhas diferentes:
  - seu valor na tabela ASCII
  - seu antecessor e o valor dele na tabela ASCII
  - seu sucessor e o valor dele na tabela ASCII

Use as funções `getche()` e `getch()` – duas opções

# Exercício

```
#include <stdio.h>

int main()
{
    char c;
    printf("Digite um caractere: ");

    c = getche();

    printf("\nValor na tabela ASCII: %d\n", c);

    printf("Caractere antecessor: %c\n", --c);
    printf("Valor do antecessor na tabela ASCII: %d\n", c);

    printf("Caractere sucessor: %c\n", c+2);
    printf("Valor do sucessor na tabela ASCII: %d\n", c+2);
}
```

# A função `getchar`

- A função `getchar()` lê o primeiro caractere de um string e termina quando a tecla <enter> for pressionada
- Ao invés de usar:

```
scanf ("%c", &c)
```

Usa-se:

```
c = getchar()
```

# Exercício

- Escreva um programa que peça ao usuário para escrever uma string na tela e que escreva o primeiro caractere da string na linha seguinte

```
#include <stdio.h>

int main()
{
    char c;
    printf("Escreva uma string: ");

    c = getchar();

    printf("\nPrimeiro caractere da string: ");
    putchar(c);
}
```

**Engenharia da Computação**

[www.eComp.Poli.br](http://www.eComp.Poli.br)

# Conceitos Fundamentais de Programação Imperativa

Disciplina: **DCExt Programação Imperativa**

Prof. Hemir Santiago

[hcs2@poli.br](mailto:hcs2@poli.br)

Material cedido pelo Prof. Joabe Jesus