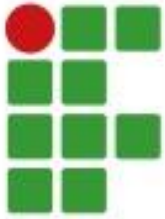

INSTITUTO FEDERAL
São Paulo
Câmpus Campinas

Linguagem de Programação I
Aula 02 – Bibliotecas, Variáveis e Saída

Profa. Cecília Sosa

cecilia.sosa@ifsp.edu.br

Profa. Joice Mendes

joice.mendes@ifsp.edu.br

INSTITUTO FEDERAL
São Paulo
Câmpus Campinas

Aula de Hoje

- Bibliotecas.
- Tipos de dados primitivos em C.
- Variáveis
  - Declaração
  - Atribuição de valores
- Saída de Dados

Bibliotecas

- Possibilitam a utilização de funções já definidas.
- Cada biblioteca fornece um conjunto de funções.
- Possuem a terminação ".h" → header.
- São iniciadas no início do arquivo fonte, através da diretiva *#include* <*nome_bib.h*>.
- As bibliotecas, bem como suas funções, possuem formas bem delimitadas.

Bibliotecas

- Algumas bibliotecas da linguagem C.

stdio.h	Funções de entrada e saída printf() / scanf()
stdlib.h	Chamadas ao sistema system(“pause”) / system(“cls”)
math.h	Funções matemáticas pow() / sqrt()
locale.h	Idiomas e acentuação setLocale()

Tipos de dados primitivos em C

- Tipos primitivos são tipos pré-definidos na linguagem.
- Os valores armazenados no processamento podem ser de diversos tipos.
- Os tipos definem os valores aceitos e as operações que podem ser realizadas.

INSTITUTO FEDERAL
São Paulo
Câmpus Campinas

Tipos de dados primitivos em C

Palavra chave	Tipo	Exemplo de valor
char	1 caractere	‘a’, ‘G’, ‘5’, ‘$’
int	Valor inteiro, sem parte decimal.	5, 87, 10
float	Real de precisão simples	1.25, 3.14
double	Real de dupla precisão.	0,11259654785522
void	Vazio – sem valor.	-

Tipos de dados primitivos em C

Tipo	Tamanho	Menor valor	Maior valor
char	1 byte	-128	+127
unsigned char	1 byte	0	+255
short int (short)	2 bytes	-32.768	+32.767
unsigned short int	2 bytes	0	+65.535
int (*)	4 bytes	-2.147.483.648	+2.147.483.647
long int (long)	4 bytes	-2.147.483.648	+2.147.483.647
unsigned long int	4 bytes	0	+4.294.967.295
float	4 bytes	-10^{38}	$+10^{38}$
double	8 bytes	-10^{308}	$+10^{308}$

(*) depende da máquina, sendo 4 bytes para arquiteturas de 32 bits

Variáveis

- É uma posição/segmento de memória capaz de armazenar um valor.
- Pode armazenar apenas um valor em um dado momento.
- Deve ser identificado por um nome.
- Está vinculada a um tipo.
- Devem ser declaradas antes de serem utilizadas para o armazenamento (escrita, leitura, modificação).

INSTITUTO FEDERAL
São Paulo
Câmpus Campinas

Declaração de Variáveis

- É uma posição/segmento de memória capaz de armazenar um valor.
- Sintaxe para declaração de variáveis em C

  tipo nome_da_variável;

- Exemplos de declarações:
  - int idade;
  - float altura;
  - char ativo;
- A declaração de uma variável **não** a associa a valor algum.

INSTITUTO FEDERAL
São Paulo
Câmpus Campinas

Identificadores de Variáveis

- Cada variável deve possuir um nome.
- Os nomes devem ser significativos.
- Os nomes obedecem a algumas regras:
  - Não pode iniciar com algarismo.
  - Não pode conter espaços.
  - Não pode conter caracteres especiais (+, -, *, &, #, etc), exceto '_'.
  - As palavras reservadas não pode ser utilizadas como nomes.
  - Diferencia minúsculos de maiúsculos (case sensitive).
  - Devem ser escritas, na manipulação, na mesma forma como foram declaradas.

Atribuição de valores a Variáveis

- Uma variável, depois de declarada, pode ter um valor associado/armazenado nela.
- A associação é realizada através da atribuição de valores.
- O operador "=" é utilizado na operação de atribuição.
- Cada tipo de dado pode apresentar necessidades diferentes na atribuição.

Atribuição de valores a Variáveis

Tipo	Declaração	Atribuição
char	char status;	status = 'l';
int	int irmaos;	irmaos = 3;
float	float saldo;	saldo = 100.58;
double	float pi;	pi = 3.1415;
void	void???	???

Saída de dados

- Permite a exibição de mensagens ou valores de variáveis no monitor.
- A saída padrão dos programas em C é apresentada no prompt/console.
- A função printf() é utilizada para a exibição de valores e/ou mensagens fixas.

Saída de dados

- Sintaxe.

```
printf("Mensagem a ser apresentada");
```

- Tudo o que estiver entre os parênteses da função (argumento) será enviado para a tela, no formato como foi definido.
- Permite o envio de valores de variáveis de diversos tipos.

INSTITUTO FEDERAL
São Paulo
Câmpus Campinas

Saída de dados

- Sintaxe.

  printf("Valor da variável = %f", nome_variável);

- Para apresentar valores de variáveis é necessário sinalizar que qual o tipo será buscado.
  - %c char
  - %d ou %i inteiro
  - %f float
  - %lf double

INSTITUTO FEDERAL
São Paulo
Câmpus Campinas

Saída de dados

- Sintaxe.

printf("Mensagem %<tipo_1> %<tipo_2>",
<variável_1>, <variável_2>);

- Exemplos:

printf("Idade = %i e altura = %.2f", idade, altura);

printf("O status do cliente %d é %c", cliente, status);

printf("%d %c %.3f", cliente, status, valor);

Exercícios

1. Crie um programa em C com uma variável com a sua idade. Apresente esse valor na tela.
2. Crie um programa em C com uma variável contendo sua altura. Apresente esse valor na tela.
3. Crie um programa em C com uma variável contendo a letra inicial do seu nome. Apresente na tela a mensagem: *A letra inicial do meu nome é LETRA*.

Referências

- ASCENCIO, A. F. G.; CAMPOS, E. A. V.. Fundamentos da Programação de Computadores Algoritmos, Pascal, C/C++ e Java. 3. ed. São Paulo: Pearson, 2012. 587p.
- BUFFONI, S. Apostila de Algoritmo Estruturado. FIAA. 4ed, 2003.
- FORBELLONE, A. L. V.; EBERSPÄCHER, H. F. Lógica de programação: a construção de algoritmos e estruturas de dados. 3ed. Pearson, 2005.
- KERNIGHAN, B. W.; RITCHIE, D. M.. Linguagem de Programação C. Rio de Janeiro: Campus, 1989. 304p.
- Code::Blocks. Disponível em http://www.codeblocks.org/. Acesso em 08 fev. 2018.
- Dev-C++. Disponível em <https://sourceforge.net/projects/orwelldevcpp/>. Acesso em 27 out. 2020.