Amarração e subprogramas

* Sintaxe (forma)
* Semântica (significado)
* Variáveis (atributos)
* Funções (atributos)
* Constantes (atributos)
* Abstração

|  |  |
| --- | --- |
| Atributos   * Nome * Tipo * Valor * Escopo * Tempo de vida | Entidades   * Variáveis * Funções * Constantes |

* Tempo de compilação
* Tempo de execução
* Amarração estática
* Amarração dinâmica
* Unidade chamadora
* Unidade chamada
* Abstração procedimental

Exemplo:

#include <stdio.h>

int quadrado(int x) {

x = x \* x;

return (x);

}

int main() {

int x, resp;

printf(“Digite um valor:”);

scanf(“%i”, %x);

resp = quadrado(x);

printf(“%i \* %i = %i\n”, x, x, resp);

}

Exercícios:

1. Considerando o seguinte trecho de código em C, liste os tipos de amarração ocorridos para cada linha de código nos tempos de compilação e execução:

int a;

scanf (“%i”,&a);

int c = a \* 5;

c = c \* a;

a = c;

|  |  |
| --- | --- |
| Nome: a  Tipo: int  Valor: ?  Escopo: trecho  Tempo de vida: fim do trecho | Nome: c  Tipo: int  Valor: a \* 5  Escopo: trecho  Tempo de vida: fim do trecho |

2. Liste todos os tipos de amarração que ocorrem no seguinte trecho de código em C:

#define TAM 30

float x = 3.2;

char str[TAM];

gets(str);

x = x \* TAM;

|  |  |
| --- | --- |
| Nome: TAM  Tipo: int  Valor: 30  Escopo: trecho  Tempo de vida: fim do trecho | Nome: x  Tipo: float  Valor: 3.2  Escopo: trecho  Tempo de vida: fim do trecho |
| Nome: str  Tipo: char  Valor: ?  Escopo: trecho  Tempo de vida: fim do trecho |  |