

Licenciatura em Informática

Resolução Perguntas 12 a 14 do Exame Tipo

12. Escreva uma classe designada por cliente com os seguintes members:

idade (**private**)

nome (**public**)

A classe deve ter dois construtores, um construtor para inicializar somente o member idade e outro para inicializar os members idade e nome.

Resposta:

```
class cliente
{
    private:
        int idade;

    public:
        char nome[ 30 ];

        cliente ( int i, char *m )
        {
            idade = i;

            strcpy(nome, m);
        }

        cliente ( int i )
        {
            idade = i;
        }

        cliente () // construtor default
        {
        }

} // Fecha a classe cliente
```

13. Reescreva a classe cliente utilizando um método apropriado de forma a permitir obter o valor do data member private idade.

Resposta:

```
class cliente // esta parte da classe mantém-se, mas
{
    private:
        int idade;

    public:
        char nome[ 30 ];

        cliente ( int i, char *m )
        {
            idade = i;

            strcpy(nome, m);
        }

        cliente ( int i )
        {
            idade = i;
        }

        cliente () // construtor default
        {

        }

        int getIdade() // mas acrescenta-se assim este método à classe cliente
        {
            return idade;
        }

} // fecha a classe cliente
```

14. Em referência à classe criada na pergunta anterior, crie na função main dois objetos. Um objeto deve inicializar apropriadamente os data members idade e nome com o valor 20 e ISTEC, respetivamente, mas através dos construtores já implementados.

Outro objeto deve ler e imprimir adequadamente o valor do campo idade da classe, sem uso de construtores.

Resposta:

```

void main()
{
    cliente cli1(20, "ISTEC");

    cliente cli2;

    printf("\n Idade do cli2 = %d", cli2.getIdade());

}

```

Nota:

Outra pergunta tipo que pode sair em vez de uma das perguntas acima:

O seguinte excerto de programa contém erros.

Identifique-os (**não é necessário refazer o código**, somente é pedido para identificá-los).

```
#include <iostream >
```

```
#define MACRO QUADRADO(x) (x * x); // A palavra MACRO está a nais
```

```

struct cliente
{
    long num;
    char nome[30];
};

```

```

struct cliente cli;
struct cli *p;

```

O código abaixo está fora de contexto (não pertence a nenhuma função!

```

cout << endl << "Insira numero do Cliente: ";
fflush(stdin);
cin >> p.num;

```

```

cout << endl << "Insira Nome do Cliente: ";
fflush(stdin);
gets(p.nome);

```

```
cout << endl << "Dados Introduzidos: " << endl << endl;
```

```

cout << "Num do Cliente: " << p.num << endl;
cout << "Nome do Cliente: " << p.nome << endl;

```

```

int void main()
{
    struct cliente cli;
    &cli;

    float lado;
    cout << "\nDigite o comprimento do lado do quadrado: ";

    cin << lado;

    cout << "O Quadrado com o lado " << lado
        << " = " << quadrado(lado) << endl;

```

Código abaixo é absurdo: Trata-se de uma função com o mesmo nome de uma macro!!!!

E sendo uma função não pode estar inserida dentro de outra função seja qual for.

```

int QUADRADO (double float x)
{
    return x * x;
}

system("pause");

} // Fecha main

```