

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ - Campus de Curitiba (Brasil) -  
Departamento Acadêmico de Eletrônica (DAELN). Disciplina: Fundamentos de Programação 2-  
IF62C/Turma: S11. Prof: Jean M. Simão. Curso: Engenharia Industrial Elétrica, ênfase  
Eletrônica/Telecomunicações. Prova sobre linguagem C++ – Prova da 1ª Parcial.**

Nome do Aluno: \_\_\_\_\_  
 Horário de Começo: \_\_\_\_\_ Horário de Fim: \_\_\_\_\_

Leia toda a prova antes de começar, pois os enunciados estão completados uns nos outros.

**(Questão 1)** Transforme o programa em ling. C abaixo em um programa C++ orientado a objetos, onde os dados de equipamentos e seus tratamentos sejam programados em uma classe *Equip*. Os atributos desta devem ser protegidos e corretamente iniciados quando da criação de objetos, por mecanismo apropriado. Certamente, deverá existir uma classe principal nesta solução.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct Equip
{
    char nome [150];
    int ano_revisao, temp_md_falha;
};

void QuandoFalhara ( struct Equip* equip, int ano_atual ) {
    int sub = ano_atual - ( equip->ano_revisao );
    int tempo = ( equip->temp_md_falha ) - sub ;
    printf ( " O %s está com ", equip->nome );
    ( tempo <= 0 ) ? printf ( "falha iminente.\n" ) : printf ( "falha provável em %d ano(s). \n", tempo);
}

void main () {
    struct Equip robo, fresa;
    robo.ano_revisao = 2006; robo.temp_md_falha = 3; strcpy ( robo.nome, "robô manipulador" );
    fresa.ano_revisao = 2007; fresa.temp_md_falha = 5; strcpy ( fresa.nome, "fresa multifuncional" );

    QuandoFalhara ( &robo, 2008 );    QuandoFalhara ( &fresa, 2008 );    system ( "pause" );
}
```

Obs.: Utilizar *cout/cin* para entrada/saída, bem como *string* em vez de vetores de caractere ordinários.

**(Questão - 2)** Elabore uma classe *Carro*, derivada de *Equip* da questão 1. Esta classe *Carro* terá dois atributos privados chamado *chassi* e *placa* que deverão (nos seus objetos) ser acessíveis de alguma maneira, por algum método. A classe *Carro* terá também o operador de igual (==) sobrecarregado, de maneira que permita comparar os atributos *ano\_revisao* de dois objetos desta classe. Ainda, instancie dois objetos da classe *Carro* na classe *Principal* comparando-os por meio do operador de igual sobrecarregado.

**(Questão - 3)** Elabore uma classe genérica ou gabarito chamada *Elemento*. Esta classe será útil para compor listas simplesmente encadeadas relativas a (ponteiros de) objetos de uma dada classe.

**(Questão - 4)** Elabore uma classe *Lista\_Carros* para (ponteiros de) objetos da classe *Carro*. Cada objeto (potencial) da classe *Lista\_Carros* pode ter um número variável de elementos inclusos. Salienta-se ainda que cada objeto *Carro* pode participar de diversas listas. Assim sendo, utilize a classe *Elemento<Tipo>* (previamente elaborada) na implementação desta classe *Lista\_Carros*.

Obs.: (a) Utilizar alocação dinâmica de memória para implementar a lista encadeada. (b) A classe *Lista\_Carros* deverá ter um método para listar bem como outro para incluir.

**(Questão - 5)** Faça com que a classe *Principal* tenha (um atributo privado que será) um objeto da classe *Lista\_Carros*. A classe *Principal* deverá ter ainda um método para listar os *Carros*, bem como outro para incluí-los. Estes métodos chamarão métodos do objeto de *Lista\_Carros* passando os parâmetros necessários (quando necessários).