

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ (UTFPR)

Campus de Curitiba-PR (Brasil) – DAELN - Departamento Acadêmico de Eletrônica.

Curso: **Engenharia Industrial Elétrica, ênfase Eletrônica/Telecomunicações.**

Disciplina: **Fundamentos de Programação 2 – IF62C – Turma: S11.**

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 C abaixo em um programa C++ orientado a objetos, onde:

- os atributos e métodos para tratar dados de pessoas sejam programados em uma classe Pessoa.
- os atributos devem ser protegidos e os respectivos métodos criados de forma apropriada.
- sobrecarregue o método de cálculo de idade com um parâmetro referente ao ano de nascimento.
- ‘sobrecarregue’ o operador *igual* em *Pessoa* para comparar os atributos de 2 objetos desta classe.
- o atributo nome não será um vetor de caracteres elementar, mas sim uma *string* do C++.

Obs.: Na verdade, não deve haver vetores de caracteres elementares nas respostas da prova, deve-se usar *strings* C++.

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

```
struct Pessoa
{
    char nome [100];
    int dia; int mes; int ano; int idade;
};

void Calc_Idade ( struct Pessoa* p, int dia, int mes, int ano )
{
    p->idade = ano - p->ano;
    if (
        ( p->mes > mes ) ||
        ( ( p->mes == mes ) && ( p->dia > dia ) )
    )
    {
        p->idade = p->idade - 1;
    }
    printf ( "A idade de %s é/seria %d \n", p->nome, p->idade );
}

int main ()
{
    struct Pessoa Einstein, Newton;
    strcpy ( Einstein.nome, "Albert Einstein" );
    strcpy ( Newton.nome, "Isaac Newton" );

    Einstein.dia = 14;   Einstein.mes = 3;   Einstein.ano = 1879;
    Newton.dia = 4;     Newton.mes = 1;   Newton.ano = 1643;

    Calc_Idade ( &Einstein, 21, 11, 2007 );
    Calc_Idade ( &Newton, 21, 11, 2007 );

    getchar ();
    return 0;
}
```

(Questão - 2) Resolva os seguintes itens:

- Elabore uma classe *Aluno* (derivada da classe *Pessoa*) e uma classe *Disciplina*.
- A classe *Disciplina*, deve ter uma lista para ponteiros de objetos da classe *Aluno*.
- Esta lista deve ser criada a partir da STL (*Standard Template Library*) do C++.
- A lista não deve passar de um número dado de elementos cujo valor padrão será 120 alunos.
- A classe *Disciplina* deve ter um método para listar os *Alunos* relacionados e outro para incluí-los.

(Questão - 3) Resolva os seguintes itens:

- Escreva um método na classe *Pessoa*, chamado de *setIdentificacao*, que sirva para armazenar o valor de RG (Registro Geral) e outro *getIdentificacao* que sirva para o informar.
- Na classe *Aluno*, faça com que o *setIdentificacao* seja um método que sirva para armazenar o valor do RG e do RA (Registro Acadêmico) do aluno, enquanto *getIdentificacao* sirva para os informar.
- Crie ainda uma classe *Professor* derivada de *Pessoa* onde o *setIdentificacao* é um método que serve para armazenar o valor do RG e do RP (Registro Profissional) do professor, enquanto *getIdentificacao* serve para os informar.
- No que diz respeito à *setIdentificacao* e *getIdentificacao*, as classes *Aluno*, *Professor* e *Pessoa* devem ser polimórficas.

Obs.: Lembre-se que em polimorfismo os métodos têm mesma assinatura e há um deles, em especial, que deve ser virtual.

(Questão – 4) Certamente, há uma classe *Principal* no programa constituído pelas soluções das questões anteriores. Assim sendo:

- Na classe *Principal*, crie uma lista para objetos da classe *Pessoa*, via STL, que armazenará apontamentos para objetos criados a partir das classes *Aluno* e/ou *Professor*.
- Na classe *Principal*, elabore ainda um método capaz de imprimir os valores do RG/RA ou RG/RP (cf. o caso) a partir dos objetos recuperados desta lista em questão.

(Questão – 5) Responda objetivamente as perguntas:

- A – O que é um método *const*?
O que é um retorno de método *const*?
O que é um atributo *const*?
- B – Explique, em linhas gerais, como se dá uma programação visual com formulários (*forms*) que suportam objetos gráficos preexistentes e tratamento de eventos.

Em programação visual, é correto implementar as funcionalidades do sistema diretamente nos métodos que tratam dos eventos sobre os objetos gráficos? Justifique sua resposta.
- C – O que são *Threads*?
- D – O que são *APIs*?
- E – O que são padrões e para que servem?