Data: 20/01/2014 – 2º Semestre de 2013.
Prof. Dr. Jean M. Simão - DAELN/UTFPR Curitiba-Central.

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ - Campus de Curitiba Central – **Dep. Acadêmico de Eletrônica** (DAELN). Disciplina: **Fundamentos de Programação 2 - IF62C**. **Prof**: Jean M. **Simão, Dr**. **Prova sobre linguagem C++ / Orientação a Objetos – 1ª Parcial**

Nome do Aluno: _____ **Turma**: _____
Curso:_____ **Horário de Começo**: _____ **Horário de Fim**: _____

Leia toda a prova, pois os enunciados estão completados uns nos outros.
Utilize os bons princípios de projeto e programação orientada a objetos.

(Questão - 1) Em um programa C++ (para *console*), crie uma classe *Componente* com um método "*virtual void informe_numero() const { cout << "O id é: " << id << endl; }*". Entretanto, esta classe pode ter outros métodos. Ainda, esta classe terá um atributo privado inteiro chamado *id*.

(Questão - 2) (2.1) Crie uma classe *Lista_Componentes* para ponteiros de objetos de *Componente*. Na prática, a classe *Lista_Componentes* servirá para ponteiros de objetos de subclasses de *Componentes* (apontados cada qual como *Componente* bem entendido). Ainda, a classe *Lista_Componentes* deverá ter um método *void incluir (Componente* pc)*, bem como outro *void listar()*. Por fim, em cada eventual objeto de *Lista_Componentes*, a listagem apresentará o '*numero*' de cada objeto cujo apontamento esteja registrado nesta lista. **(2.2)** Esta classe *Lista_Componentes* deve ser baseada em *STL* agregando, por exemplo, um objeto da classe *Vector* ou *List* parametrizada com (o tipo) ponteiro de *Componente*. Alternativamente, a classe *Lista_Componentes* pode ser uma lista encadeada feita pelo desenvolvedor baseada em *template* ou mesmo específica.

(Questão - 3) (3.1) Crie uma classe *Roda* (com inteiro chamado *aro*), uma classe *Chassi* (com um atributo string chamado *num_chassi*) e uma classe *Lataria* sendo todas derivadas de *Componente*. **(3.2)** As classes *Roda* e *Chassi* devem redefinir o método chamado "*virtual void informe_numero() const*". No caso de *Roda* tal método informará (em tela) "*Aro da roda:"* seguido do valor do seu atributo *aro*. Ainda, no caso de *Chassi* tal método informará (em tela) "*Chassi de número:"* seguido do valor seu atributo *numero*. **(3.3)** Outrossim, o operador de fluxo de saída *operator<<* será sobrecarregado para a classe *Chassi* para permitir mostrar em tela o valor de atributos *numero* e *id* de objeto desta classe. **(3.4)** Por fim, a classe *Chassi* permitirá apontamentos para quatro objetos da classe *Roda* (podendo utilizar aqui um objeto de *Lista_Componentes* para tal) e a classe *Lataria* permitirá um apontamento para um objeto de classe *Chassi* (idem).

(Questão - 4) Crie uma classe *Carro* sendo que cada eventual objeto dela poderá armazenar em si ponteiros de objetos da classe *Roda*, da classe *Chassi* e da classe *Lataria* por meio de um método *void incluir_componente (Componente* pe)*. Assim, visando polimorfismo, cada eventual objeto da classe *Carro* armazenará os apontamentos de objetos da classe *Roda*, *Chassi* e *Lataria* como apontamentos de objetos da classe *Componente*. Isto se dará por meio de um objeto agregado do tipo *Lista_Componentes*. Por fim, a classe *Carro* terá um método *void listar()* que chamará método apropriado deste objeto de lista agregado.

(Questão - 5) (5.1) Crie uma classe *Principal* onde será criado dois objetos da classe *Carro* (*Fusca_1* e *Kombi_1*), bem como respectivos objetos da classe *Roda*, *Chassi e Lataria*. Certamente, os objetos da classe *Roda*, *Chassi e Lataria* serão apropriadamente associados entre si e também ao objeto de *Carro* no método construtor da classe *Principal*. Ainda, a classe *Principal* terá um método *void executar()* que permitirá listar o *número* dos objetos considerados em cada objeto da classe *Carro*. Por fim, a classe *Principal* será instanciada na função *main()* do programa. **(5.2)** Faça diagrama de classes das classes e relacionamentos destas conforme solicitado nesta questão e nas precedentes.

A interpretação faz parte do conteúdo da prova! Inclua comentários (se for o caso) para deixar explicitas suas decisões.