**1) Preencha as seguintes lacunas:**

a) Os membros de classe são acessados via operador **.** em conjunto com o nome de um objeto (ou

referência a um objeto) da classe ou via operador **->** em conjunto com um ponteiro para um objeto da classe.

b) Os membros de classe especificados como **private** são acessíveis às funções-membro da classe e friend da classe.

c) Os membros de classe especificados como **public** são acessíveis em qualquer lugar que um objeto da classe esteja no escopo.

d) A **construção de cópia(?)** pode ser utilizada para atribuir um objeto de uma classe a outro objeto da mesma classe.

e) Ao separar a interface de uma classe da sua implementação, o link entre os métodos é feito através do nome da classe seguido do operador **::** .

f) Um função não-membro deve ser declarada como **friend** de uma classe para ter acesso aos membros private dessa classe.

g) Toda sobrecarga de operador merece uma atenção especial quanto ao seu retorno e a sobrecarga é feita através da palavra chave **operator**.

h) A toda classe deve-se definir explicitamente os 3 construtores, o destrutor e a sobrecarga do operador de **atribuição (=)**.

i) A **herança** é uma forma de reutilização de software em que novas classes absorvem os dados e comportamentos de classes existentes aprimorando-as com novas capacidades.

j) Ao derivar de uma classe básica pela herança public, membros public se tornam **public** na classe derivada, membros protected se tornam **protected** na classe derivada e membros **private** não são herdados diretamente.

**2) Descreva de forma breve cada um dos construtores e o destrutor, dizendo os detalhes de sua implementação e em qual momento eles são acionados.**

**RESPOSTA**:

**Construtor padrão (default): é um método da classe que não recebe parâmetros e inicializa os atributos do objeto com algum valor padrão. Sua implementação é feita utilizando simplesmente o nome da classe, sem nenhum parâmetro [Ex. "Nome\_da\_classe();"]. Este construtor é acionado na inicialização de um objeto da classe quando este não recebe nenhum valor predefinido.**

**Construtor normal: é um método da classe que recebe parâmetros e inicializa os atributos do objeto de acordo com o valor dos parâmetros informados. Este construtor pode atuar como um construtor-padrão fazendo com que seus parâmetros na definição tenham valores iniciais. Sua implementação é feita utilizando o nome da classe, seguido dos parâmetros que serão utilzados [Ex. "Nome\_da\_classe(int, char, double);"]. Este construtor é acionado na inicialização de um objeto da classe quando é informado um valor predefinido.**

**Construtor de cópia: é um método da classe que recebe parâmetros e cria uma cópia bit a bit do objeto que está sendo passado como parâmetro. Sua implementação é feita utilizando o nome da classe, seguido dos parâmetros que devem conter a palavra chave "const" seguido do nome da classe, do operador '&' e do objeto que será copiado [Ex. "Nome\_da\_classe(const Nome\_da\_classe &obj);"]. Este construtor é acionado na inicialização de um objeto da classe a partir de outro objeto da mesma classe.**

**Destrutor: é um método da classe que não recebe parâmetros e realiza a destruição de um objeto. Sua implementação é feita utilizando o nome da classe antecedido por um '~' [Ex. "~Nome\_da\_classe();"]. Destrutores são acionados pelos objetos no fim do escopo em que estavam presentes, ou em alocação dinâmica quando a respectiva área é desalocada.**

**3) Descreva de forma breve quais modificações são feitas quando construtores e destrutores são implementados na hierarquia de herança, e como é a sua forma de acionamento neste caso.**

**RESPOSTA**:

**Na instanciação de um objeto da classe derivada, primeiramente são chamados os construtores da classe básica:**

**● explicitamente via inicializadores de membro**

**● implicitamente quando são construtores-padrão**

**Na destruição de um objeto da classe derivada, o destrutor da classe básica é chamado implicitamente.**

**Construtores de Cópia e Operadores de Atribuição sobrecarregados não são herdados por classes derivadas. Entretanto, os construtores de cópia e operadores de atribuição sobrecarregados de uma classe derivada devem chamar explicitamente os respectivos da classe básica.**

**Neste caso ocorre automaticamente uma conversão do tipo derivado para o tipo básico.**