**Nome :** Ezequiel Teotônio Jó

**Lista de Exercícios – LP1 – Prof. Thiago Maritan**

**1 –** Classes são conjuntos de variáveis e métodos que servem como molde para os que os Objetos sejam criados, mas também são vistas como tipos definidos pelo programador. Objetos são abstrações da percepção humana sobre os objetos reais, aproximando cada vez mais a programação ao alto nível. Exemplo: na classe Automóvel podemos definir vários objetos que tenham as características da classe em comum, como um objeto Carro, um objeto Caminhão, etc.

**2** – O Construtor é utilizado no momento que um Objeto será criado, e tem a função de inicializar e garantir que essa inicialização do respectivo Objeto ocorreu de forma correta. Ele também é utilizado para definir as variáveis iniciais do Objeto em questão.

**3** – O *new* é um recurso utilizado na linguagem C++ para indicar uma alocação dinâmica de memória. Essa alocação funciona quando queremos criar uma variável mas não sabemos ao certo qual o espaço na memoria que ela irá ocupar, como o programa é executado em tempo de compilação, ele será abortado se inicializarmos uma variável sem definirmos o seu tamanho. Com a alocação dinâmica de memória, por exemplo, um array é declarado e futuramente é possivel definir ou alterar o seu tamanho pois o operador *new* retorna o endereço onde começa o bloco de memória que foi reservado, possibilitando o seu uso num ponteiro para alterar o conteúdo da memória alocada.

**4** – O encapsulamento traz para a POO mais segurança aos seus métodos e variáveis restringindo o seu acesso por meio de níveis de encapsulamento, que são eles: *public* ( os dados podem ser acessados por qualquer classe), *protected* (somente pode ser acessado pela classe e suas sub-classes) e *private* (apenas pode ser acessado na própria classe). Exemplo: utilizando-se do encapsulamento num sistema bancário, é possível restringir o acesso a variáveis como saldo da conta, definindo essa variável com o nível *private*, assim garantindo que esse valor não será alterado de forma indevida.