

## SÉTIMA LISTA DE EXERCÍCIOS DE PROGRAMAÇÃO III

- 1) Em que situações deve-se utilizar o operador “->” e em que situações deve-se utilizar o operador “.” Em C++? Exemplifique.
- 2) Em relação a sequência de heranças abaixo, com diferentes modificadores de acesso, informe quais membros (atributos e métodos) a classe CarroSport terá acesso e com qual modificador de acesso.

```
class MeioTransporte{
public:
    int anoFabricacao;
private:
    void imprimeAno() {}
};

class MeioTransporteTerrestre: private MeioTransporte{
protected:
    int numEixos;
public:
    void imprimeNumEixos() {}
    ~MeioTransporteTerrestre() {}
};

class Veiculo: public MeioTransporteTerrestre{
protected:
    string placa;
    void imprimePlaca() {}
public:
    Veiculo() {}
};

class Carro: protected Veiculo{
private:
    string marca;
    string modelo;
protected:
    void imprimeMarcaModelo() {}
    Carro(string _marca, string _modelo) {}
    ~Carro() {}
public:
    friend imprimeInfoCarro();
};

class CarroSport : public Carro {
private:
    int potencia;
    double valor;
public:
    void imprimePotencia() {}
};
```

- 3) O que são Exceções? Quais são as palavras chaves utilizadas para manipular uma exceção e qual a função de cada uma? Como elas podem ser tratadas? Como elas podem ser lançadas pelo próprio programador? Implemente um código exemplificando o tratamento de exceções.
- 4) O que são templates em C++? Implemente um código que exemplifique um Template para Função e um código que exemplifique um Template para Classe.
- 5) Em uma herança é possível e correto chamar os métodos da superclasse dentro dos métodos da subclasse em C++ e Java? Para cada linguagem, explique e dê exemplos.