

**UNIVERSIDADE DE AVEIRO**  
DEPARTAMENTO DE ELECTRÓNICA E TELECOMUNICAÇÕES  
**Exame Teórico de Paradigmas de Programação I**  
6/Fevereiro/2002 (duração uma hora).

I. Que operadores serão chamados em cada uma das linhas seguintes?

```
unsigned vect[]={0x1234, 0x5678}, vect1[]={0x1234, 0x9a};
B_vector BV1(40,vect), BV2(48,vect1);
BV1 = 0xabcd;
BV1 = vect;
BV1 += BV2;
BV1[i] = vect1[i];
vect[i] = BV1[i];
BV1 = 0xABCD & BV1;
cout << (BV1 | 0x1F1A) << endl;
cout << (BV1 >> 10) << endl;
cin >> BV1 >> BV1[i] >> endl;
int i = (int)BV1;
int k = (int)BV1(0);
int j = (int)BV1[0];
double *d=new(5) double;
cout << BV1(1,2) << endl;
```

**exemplo: BV1++; resposta: a função membro da classe B\_vector operator++(int);**  
**Valor 0-4.**

II. O programa seguinte tem alguns erros. Marque as linhas com erros e para cada uma delas apresente todas as correcções necessárias.

```
class my_static {
public:
    static b;
    int c = 15;
    static void f(void)
    { cout << a << b << c << endl; }
private:
    static a;
};

int my_static::a = 5;
int my_static::b = 10;

int main(int argc, char* argv[])
{ my_static::f();
  return 0; }
```

**Valor 0-3.**

III. Definir um manipulador de formatos mp(n,m,c) novo com três argumentos que são:

n - largura;  
m - precisão;

c - caracter para preencher espaços

**Valor 0-4.**

IV. O programa seguinte tem alguns erros. Marque as linhas com erros e para cada uma delas apresente todas as correções necessárias.

```
class Base
{
    public:
        void function1() const { a=10; cout << a << endl; }
        virtual void function() = 0;
        Base();
        virtual ~Base();
    protected:
        static int a;
};

class Derivada : public Base
{
    public:
        void function() { cout << "função_derivada\n"; }
        Derivada();
        virtual ~Derivada();
};

int main(int argc, char* argv[])
{
    Base b1;
    Base *b2;
    Derivada &d1=b1;
    Derivada d2;
    const Derivada d3;
    b2=&d2;
    b2->function();
    d3.function();
    d3.function1();
    return 0;
}
```

**Valor 0-4.**

V. Relativamente a cada uma das seguintes afirmações sobre a linguagem C++, indique (à esquerda da numeração) se é verdadeira (**V**) ou falsa (**F**).

1. O construtor de cópia de uma classe derivada deve chamar o construtor por defeito da classe base
2. O destrutor de uma classe não pode ser chamado
3. Um construtor não pode ter parâmetros com valores por defeito
4. Um construtor pode ser declarado como virtual

5. Todos os operadores do C++ podem ser redefinidos
6. O operador `()` é considerado um operador binário
7. Com exceção do operador `=`, todos os operadores redefinidos na classe B são herdados por qualquer classe D derivada de B
8. O operador global delete pode ser redefinido
9. Na redefinição de operadores podem ser usados argumentos por defeito
10. Um ponteiro para uma classe B pode ser atribuído a uma variável do tipo ponteiro para classe D sem ser necessária uma conversão explícita (D é derivada de B)
11. As funções virtuais podem ser estáticas
12. Uma classe pode ser declarada como classe base mais do que uma vez
13. A declaração de template só pode ser global
14. As exceções podem ser de qualquer tipo
15. Uma função com uma lista de especificação de exceções vazia, tal como `throw()`, não pode gerar qualquer exceção
16. A procura dos blocos catch é feita pela sequência em que estes blocos se encontram no programa
17. A classe pessoa é uma generalização da classe aluno
18. Um manipulador de formato (para entrada e saída de dados) é válido desde o seu aparecimento do manipulador até ao aparecimento de outro manipulador
19. Uma classe derivada pode ter as mesmas classes base virtuais e não virtuais herdadas através de classes intermédias
20. Uma classe para representar um **aluno** pode ser derivada de uma classe que representa uma **universidade**.

O valor para cada pergunta pode ser -0.25 (a resposta é errada), 0 (não há nenhuma resposta) 0.25 (a resposta é correcta).

**Valor 0-5.**