

Instituto Superior de Engenharia de Coimbra

Programação II

Exame da época normal

3-12-2007

Nome: _____

Número: _____

1 – Qual será a saída resultante da execução deste programa? (escolha uma hipótese)

```
class A {
public:
    virtual void f(){ cout << "f de A \n"; }
    void g(){ cout << "g de A \n"; }
};
class B: public A {
public:
    virtual void f(){ cout << "f de B \n"; }
    void g(){ cout << "g de B \n"; }
};

void listar( A * a){
    a->f();
    a->g();
}

void main(){
    listar(new A);
    listar(new B);
}
```

a)

```
f de A
g de A
f de B
g de A
```

b)

```
f de A
g de A
f de A
g de A
```

c)

```
f de A
g de A
f de B
g de B
```

d)

```
f de A
g de A
f de A
f de B
g de A
```

2 – Qual será a saída resultante da execução deste programa? (escolha uma hipótese)

```
class A {
public:
    ~A(){ cout << "~A "; }
};
class B: public A {
public:
    ~B(){ cout << "~B "; }
};

class X {
public:
    virtual ~X(){ cout << "~X "; }
};
class Y: public X {
public:
    virtual ~Y(){ cout << "~Y "; }
};
```

```
void main(){
    A * a = new B;
    delete a;

    X * x = new Y;
    delete x;
}
```

a) ~B ~Y

b) ~A ~Y ~X

c) ~A ~B ~X ~Y

d) ~B ~A ~Y ~X

3 – Qual será a saída resultante da execução deste programa? (escolha uma hipótese)

```
class A {
    int n;
public:
    A(int nn = 0){ n = nn; }
    A( const A & ob){ n = ob.n; }
    void listar(){ cout << " n= " << n; }
};
class B: public A {
    int k;
public:
    B(int nn = 0, int kk = 0):A(nn){ k = kk; }
    B( const B & ob){ k = ob.k; }
    void listar(){
        A::listar();
        cout << " k= " << k << endl;
    }
};
void main(){
    B b(4, 6);
    b.listar();
    B x(b);
    x.listar();
}
```

a)

n = 4 k = 6
n = 4 k = 6

b)

n = 4 k = 6
n = 0 k = 6

c)

n = 0 k = 6
n = 0 k = 6

d) Erro porque um construtor de uma classe derivada tem que invocar explicitamente um construtor da classe base.

4 – Qual será a saída resultante da execução deste programa? (escolha uma hipótese)

```
class A {
    // ...
};

void main(){
    A x(2), y(3), z;
    z = x + y;
}
```

Entre as funções seguintes, qual a função que necessita de estar definida para que este código compile sem erros. (escolha uma hipótese)

a) A operator+(const A & ob) ou A A::operator+(const A & ob1, const A & ob2)

b) void A:: operator+(const A & ob) ou void operator+(const A & ob1, const A & ob2)

c) A A:: operator+(const A & ob) ou A operator+(const A & ob1, const A & ob2)

d) A A:: operator+=(const A & ob)

5 – Considere as seguintes definições das classes Num e ListaDeNumeros:

```
class Num{
    int n;
public:
    Num(int nn):n(nn){}
    void listar(){ cout << " n= " << n << endl; }
};

class ListaDeNumeros {
    vector <Num *> numeros;
public:
    void acrescentaNumero1( int k){
        Num x(k);
        numeros.push_back(&x);
    }
    void acrescentaNumero2( Num *p){
        numeros.push_back(*p);
    }
    void acrescentaNumero3( int k){
        Num * p = new Num(k);
        numeros.push_back(p);
    }
    void acrescentaNumero4( Num ob){
        Num * p = new Num(ob);
        numeros.push_back(p);
    }
};
```

Pretende-se uma função que acrescente um *num* a esta *lista de números*.

Quais das funções membros da classe ListaDeNumeros cumpririam correctamente esse objectivo? (escolha uma hipótese)

- a) acrescentaNumero1() e acrescentaNumero3()
- b) acrescentaNumero3() e acrescentaNumero4()**
- c) Apenas acrescentaNumero3(), todas as outras estão incorrectas
- d) acrescentaNumero2() e acrescentaNumero4()

6 – Dado o seguinte programa:

```
class Numeros {
    int a[10];
public:
    Numeros(){
        for(int i = 0 ; i < 10 ; i++)
            a[i] = 9;
    }
    int * getA(){ return a; } // linha 8
    const int * getAA ()const{ return a; } // linha 9
};

void main(){
    Numeros x;
    x.getA()[0] = 4; // linha 13
    x.getAA()[1] = 4; // linha 14
    x.listar();
}
```

Indique quais as linhas deste programa em que ocorrem erros de compilação (escolha uma hipótese).

- a) 9, 13, 14
- b) 13, 14
- c) 14**
- d) Não tem erros de compilação

7 – Dado o seguinte programa:

```
class A{
public:
    virtual void f()=0;
    virtual void g()=0;
    virtual void h(){}
};
class B: public A{
public:
    virtual void g(){}
};
class C: public B{
public:
    virtual void f(){}
};
void main(){
    A a;    // linha 1
    B b;    // linha 2
    C c;    // linha 3

    A * p = & c; // linha 4
    A & ref = c; // linha 5

    p->f();    // linha 6
    p->g();    // linha 7
    p->h();    // linha 8

    ref.f();   // linha 9
    ref.g();   // linha 10
    ref.h();   // linha 11
}
```

Indique quais as linhas deste programa em que ocorrem erros de compilação (escolha uma hipótese).

a) 1, 2, 5, 9, 10, 11

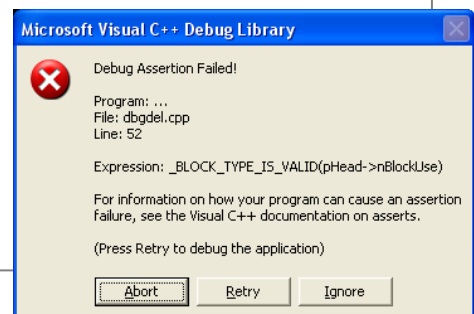
b) Não há erros

c) 7, 8, 10, 11

d) 1, 2

8 – Ao executar este programa,

```
class Palavra {
    char * p; // ponteiro para array dinâmico de caracteres
public:
    Palavra( char * s){
        if (!s){
            p = 0;
        } else {
            p = new char[strlen(s)+1];
            strcpy(p,s);
        }
    }
    ~Palavra(){ delete [] p;}
};
void main(){
    Palavra x("abc");
    Palavra y("xyz");
    y = x;
}
```



aparece a seguinte indicação de erro:

Este erro deixa de existir se for devidamente definida uma das seguintes funções ? (escolha uma hipótese)

a) Palavra &Palavra::operator=(const Palavra & ob);

b) Palavra::Palavra(const Palavra & ob);

c) ostream & operator<<(ostream & saida, const Palavra & ob);

d) bool operator<(const Palavra & a, const Palavra & b);

9 – Qual será a saída resultante da execução deste programa? (escolha uma hipótese):

```
class Produto{
    static int n;
public:
    Produto(){}
    Produto( const Produto & ob){ ++n ; }
    static int getN(){ return n;}
};
int Produto::n = 0;
void f( Produto & ob){}
void g( Produto ob){}
void main(){
    cout << "\nA: " << Produto::getN();
    Produto a;
    Produto b(a);
    cout << "\nB: " << Produto::getN();
    f(a);
    cout << "\nC: " << Produto::getN();
    g(a);
    cout << "\nD: " << Produto::getN();
}
```

a)

A: 0
B: 1
C: 2
D: 3

b)

A: 0
B: 0
C: 0
D: 0

c)

A: 3
B: 3
C: 3
D: 3

d)

A: 0
B: 1
C: 1
D: 2

10 – Qual será a saída resultante da execução deste programa? (escolha uma hipótese):

```
class Num{
    int n;
public:
    Num(int nn):n(nn){}
    int getN()const{ return n; }
    void setN( int nn){ n = nn;}
    void listar(){ cout << " n= " << n << endl; }
};

Num & aumenta( Num & ob){
    ob.setN(ob.getN()+1);
    return ob;
}

Num diminui( Num ob){
    ob.setN(ob.getN()-1);
    return ob;
}

void main(){
    Num a(6);
    aumenta(aumenta(a));
    a.listar();
    Num b(6);
    diminui(diminui(b));
    b.listar();
}
```

a)

n = 8
n = 4

b)

n = 6
n = 4

c)

n = 8
n = 6

d)

n = 6
n = 6