

Instituto Superior de Engenharia de Coimbra

Programação II

Exame da época normal

23-11-2007

Nome: _____

Número: _____

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

```
class A {
public:
    virtual void funcV(){ cout << "\n A: funcV "; }
    void g(){ cout << "\n A: g "; }
};
class B: public A {
public:
    virtual void funcV(){ cout << "\n B: funcV "; }
    void g(){ cout << "\n B: g "; }
};
void main(){
    A * x []= { new A, new B };
    x[0]->funcV();
    x[1]->funcV();
    x[0]->g();
    x[1]->g();
}
```

a)

A: funcV
A: funcV
A: g
A: g

b)

A: funcV
B: funcV
A: g
B: g

c)

A: funcV
B: funcV
A: g
A: g

d)

A: funcV
A: funcV
B: funcV
A: g
A: g

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

```
class A {
public:
    A(){ cout << "\n A()"; }
    A(const A & ob){ cout << "\n A(const A & ob)"; }
};
class B: public A {
public:
    B(){ cout << "\n B()"; }
    B(const B & ob){ cout << "\n B(const B & ob)"; }
};
void main(){
    B x;
    B y(x);
}
```

a)

B()
B(const B & ob)

b)

A()
B()
A()
B(const B & ob)

c)

A()
B()
A(const A & ob)
B(const B & ob)

d)

B()
A()
B(const B & ob)
A()

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

```
class Registo{
    int codigo;
public:
    Registo(){ codigo = 0; }
    void setCodigo( int c) { codigo = c; }
    int getCodigo(){ return codigo; }
};
void preenche1( Registo r){
    r.setCodigo(5);
}
void preenche2( Registo &r){
    r.setCodigo(5);
}
void preenche3( Registo *r){
    r->setCodigo(5);
}
void main(){
    Registo reg1, reg2, reg3;
    preenche1(reg1);
    preenche2(reg2);
    preenche3(&reg3);
    cout << "\n reg1: " << reg1.getCodigo();
    cout << "\n reg2: " << reg2.getCodigo();
    cout << "\n reg3: " << reg3.getCodigo();
}
```

a)

```
reg1: 5
reg2: 5
reg3: 5
```

b)

```
reg1: 0
reg2: 0
reg3: 0
```

c)

```
reg1: 0
reg2: 0
reg3: 5
```

d)

```
reg1: 0
reg2: 5
reg3: 5
```

4 – Dado o seguinte programa:

```
int & f(){
    int x;
    return x;
}
void main(){
    f()=10;
    cout << f() << endl;
}
```

Relativamente a este programa, indique qual das seguintes afirmações é verdadeira. (escolha uma hipótese)

- a) Está correcto.
- b) Está errado porque o tipo de retorno de uma função não pode ser uma referência.
- c) Está errado porque a chamada a uma função não pode estar no primeiro membro de uma atribuição.
- d) Está errado porque a função `f ()` retorna uma referência para uma variável local.

5 – Considere as seguintes definições das classes `Registro` e `Lista`:

```
class Registro{
    int codigo;
public:
    Registro(int c = 0){ codigo = c; }
    int getCodigo(){ return codigo; }
};
class Lista {
    vector<Registro *> registros;
public:
    void acrescentaRegistro( Registro * p){ if (p) registros.push_back(p);}
    void imprime(){
        for (unsigned int i = 0 ; i < registros.size() ; i++)
            cout << "\n " << i << ":" << registros[i] ->getCodigo();
    }
};
```

Pretende-se uma função que acrescente dois *registos* a esta *lista*.

Quais das seguintes funções cumpririam correctamente esse objectivo? (escolha uma hipótese)

```
void colocaRegistos1( Lista lista){
    lista.acrescentaRegistro(new Registro(11));
    lista.acrescentaRegistro(new Registro(22));
}
void colocaRegistos2( Lista &lista){
    lista.acrescentaRegistro(new Registro(11));
    lista.acrescentaRegistro(new Registro(22));
}
void colocaRegistos3( Lista lista){
    Registro um(11), dois(22);
    lista.acrescentaRegistro(&um);
    lista.acrescentaRegistro(&dois);
}
void colocaRegistos4( Lista &lista){
    Registro um(11), dois(22);
    lista.acrescentaRegistro(&um);
    lista.acrescentaRegistro(&dois);
}
```

- a) `colocaRegistos1()` e `colocaRegistos2()` b) `colocaRegistos2()` e `colocaRegistos4()`
c) `colocaRegistos2()` d) todas
e) nenhuma

6 – Considere o seguinte programa:

```
double area( double comp=1, double larg=1) {
    return comp*larg;
}
double area(double lado) {
    return lado*lado;
}
void main() {
    area(4);
}
```

Qual das seguintes afirmações é verdadeira ? (escolha uma hipótese)

- a) Não é possível inicializar os parâmetros das funções no protótipo.
b) Não é possível ter duas funções diferentes com o mesmo nome.
c) A chamada à função, `area(4)`, poderia ser executada de mais do que uma maneira, gerando assim um erro de ambiguidade.
d) Só os construtores das classes podem ter parâmetros com valores por defeito.

7 – Dado o seguinte programa:

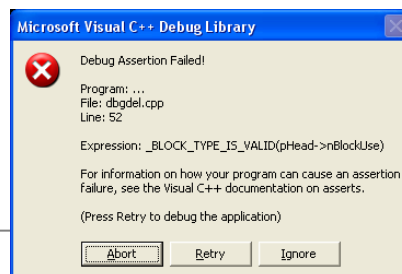
```
class Num {
    int n;
public:
    Num(){ n = 0; }
    void setN(int nn){ n = nn; }
};
void f( const Num * p){
    p->setN(5);           // linha 8
    p = new Num();       // linha 9
}
void g( Num * const p){
    p->setN(5);           // linha 12
    p = new Num();       // linha 13
}
void h(const Num * const p){ // linha 15
    p->setN(5);           // linha 16
    p = new Num();       // linha 17
}
void main() {
    Num x;
    f(&x);
    g(&x);
    h(&x);
}
```

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

- a) 8, 9, 12, 13, 16, 17 b) 8, 13, 16, 17
c) 8, 12, 16 d) 8, 9, 12, 13, 15, 16, 17

8 – Ao executar este programa,

```
class Palavra {
    char * p; // ponteiro para array dinamico de caracteres
public:
    Palavra( char * s){
        if (!s){
            p = 0;
        } else {
            p = new char[strlen(s)+1];
            strcpy(p,s);
        }
    }
    ~Palavra(){ delete [] p;}
};
void f( Palavra ob){}
void main(){
    Palavra x("abc");
    f(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 – Dado o seguinte programa:

```
class A{
public:
    virtual void f()=0;
    virtual void g(){}
};
class B: public A{
public:
    virtual void g(){}
    virtual void h(){}
};
class C: public A{
public:
    virtual void f(){}
    virtual void h(){}
};

void main(){
    A * x = new A;    // linha 1
    A * y = new B;    // linha 2
    A * z = new C;    // linha 3

    z->f();            // linha 4
    z->g();            // linha 5
    z->h();            // linha 6

    C * c = new C;    // linha 7
    c->h();            // linha 8
}
```

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

a) 1, 2, 3, 4, 5, 6

b) Não há erros

c) 1

d) 1, 2, 6

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

```
class Produto{
    static int n;
public:
    Produto(){ ++n ; }
    static int getN(){ return n;}
};
int Produto::n = 0;

void main(){
    cout << "\nA: " << Produto::getN();
    Produto a[2];
    cout << "\nB: " << Produto::getN();
    Produto * p = new Produto;
    cout << "\nC: " << Produto::getN();
}
```

a)

A:0
B:2
C:3

b)

A:0
B:0
C:0

c)

A:3
B:3
C:3

d)

A:0
B:2
C:4