Instituto Superior de Engenharia de Coimbra
Programação II

Exame da época normal 27-11-2008

Nome:_____

Número_____

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

```
class SerVivo {
public:
       string nadar(){ return string(" nadar ? "); }
       virtual string voar(){ return string(" voar ? "); }
};

class Ave: public SerVivo {
public:
       string nadar(){ return string(" pouco "); }
       virtual string voar(){ return string(" muito bem "); }
};

void main(){
       SerVivo * x[] = { new SerVivo, new Ave };
       cout   << x[0]->nadar() << endl
              << x[1]->nadar() << endl
              << x[0]->voar()  << endl
              << x[1]->voar()  << endl;
}
```

a)
```
nadar ?
nadar ?
voar ?
voar ?
```

b)
```
nadar ?
pouco
voar ?
muito bem
```

c)
```
nadar ?
pouco
voar ?
voar ?
```

d)
```
nadar ?
nadar ?
voar ?
muito bem
```

2 – Dadas as seguintes definições:

```
class Base {
       // membros da classe Base
public:
       Base(const Base & ob);
};
class Teste: public Base {
       int n;
public:
       Teste(const Teste & ob);
};
```

Como seria correcta a definição da função `Teste(const Teste & ob)` ? (escolha uma hipótese)

a) `Teste::Teste(const Teste & ob):Base(ob){  n = ob.n; }`

b) `Teste::Teste(const Teste & ob):Base(){  n = ob.n; }`

c) `Teste::Teste(const Teste & ob){ n = ob.n; }`

d) `Teste::Teste(const Teste & ob):Base(ob){  n = ob.n; }` ou

`Teste::Teste(const Teste & ob){  n = ob.n; }`

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

```
class Num {
	int n;
public:
	Num(int nn){ n = nn;}
};
class Conjunto{
	vector<Num> numeros;
public:
	vector<Num> getNumeros(){ return numeros; }
	void acrescenta( Num x){ numeros.push_back(x);}
};
void main(){
	Conjunto a;
	a.getNumeros().push_back(Num(2));
	a.getNumeros().push_back(Num(4));
	cout << a.getNumeros().size() << endl;
	a.acrescenta(Num(6));
	a.acrescenta(Num(8));
	cout << a.getNumeros().size() << endl;
}
```

a)
```
0
0
```

b)
```
0
2
```

c)
```
2
2
```

d)
```
2
4
```

4 - Considere as seguintes definições:

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

Para que a classe **C** seja concreta, isto é, para que possam existir objectos de **C**, quais as funções que é preciso implementar nesta classe ?

a) É preciso definir apenas a função `g()`.

b) É preciso definir um construtor.

c) É preciso definir as funções `f()` e `g()`. Se faltar uma destas funções, a classe **C** é abstracta.

d) Para além de definir as funções funções `f()` e `g()`, é ainda necessário definir outra função que não seja adquirida por herança.

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

```
class Base {
public:
	~Base(){ cout << "~Base \n"; }
};
class Deriv: public Base {
public:
	~Deriv(){ cout << "~Deriv \n"; }
};
void main(){
	Base * a = new Deriv;
	delete a;

	Deriv * x = new Deriv;
	delete x;
}
```

a)
~Base
~Deriv
~Base
~Deriv

b)
~Deriv
~Base
~Deriv
~Base

c)
~Base
~Deriv

d)
~Base
~Deriv
~Base

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

```
class Num {
	int n;
public:
	Num(int nn){ n = nn;}
	void setN( int nn){ n = nn;}
	int getN(){ return n;  }
};

void altera(Num a, Num & b){
	a.setN(5);
	b.setN(5);
}
void main(){
	Num x(2), y(2);
	cout << "x: " << x.getN()<< " y: " << y.getN() << endl;
	altera( x, y);
	cout << "x: " << x.getN()<< " y: " << y.getN() << endl;
}
```

a)
x: 2 y: 2
x: 2 y: 5

b)
x: 2 y: 2
x: 2 y: 2

c)
x: 2 y: 2
x: 5 y: 5

d)
x: 2 y: 2
x: 0 y: 0

7 – Dado o seguinte programa:

```
class Num {
	int n;
public:
	Num(int nn){ n = nn;}
	void setN( int nn){ n = nn;}
	int getN(){ return n;  }
	// . . .
};
void main(){
	Num a(2), b(4);
	a += b;
}
```

Quais as funções que necessitam de estar definidas para que este código não tenha erros? (escolha uma hipótese)

a) `Num & operator +=( const Num & ob)` (função global)

b) `Num & Num::operator +=( const Num & ob)` (função membro da classe Num)

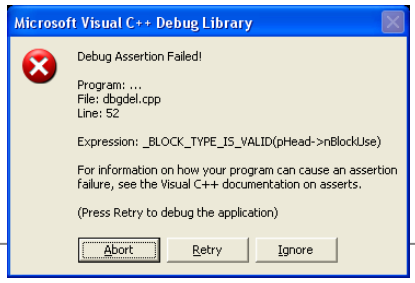
c) `Num & Num::operator =( const Num & ob)` (função membro da classe Num)

e `Num operator+( Num a, Num b)` (função global)

d) Não é preciso nenhum operador porque esta classe não tem ponteiros para memória dinâmica.

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"), y;
	f(x);
	y = x;
}
```



aparece a seguinte indicação de erro:

Este erro deixa de existir se forem devidamente definidas as 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) `Palavra::Palavra(const Palavra & ob)` e
`Palavra &Palavra:: operator=(const Palavra & ob).`

9 – Dado o seguinte programa:

```
class Figura{
public:
	virtual void area()=0;
};
class Quadrado: public Figura{
	// . . .
public:
	virtual void area();
};
void main(){
	Figura * x = new Figura;     // linha 1
	Figura * y = new Quadrado;   // linha 2
	Quadrado * z = new Quadrado; // linha 3

	y->area();    // linha 4
	z->area();    // linha 5

	y.area();     // linha 6
	z.area();     // linha 7
}
```

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

a) 1, 2, 6 e 7 **b)** 1, 2, 4 e 6

c) 1, 6 e 7 **d)** 1, 2, 4 e 5

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

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

void f(){
	Produto x, y;
	cout << Produto::getN() << endl;
}
void main(){
	cout << Produto::getN() << endl;
	f();
	cout << Produto::getN() << endl;
}
```

a)
```
0
2
0
```

b)
```
0
0
0
```

c)
```
0
2
2
```

d)
```
0
1
0
```