

		<b>Escola de Engenharia da Universidade do Minho</b> Mestrado Integrado em Eng. Electrónica Industrial e Computadores <b>Complementos de Programação de Computadores</b>						<b>2014/2015</b> <b>MIEEIC</b> <b>(1º Ano)</b> <b>2º Sem</b>	
<b>Docentes: Luís Paulo Reis, Luís Magalhães, Ricardo Carrola</b>									
<b>Exame Nº1 - Época Normal - Data 25/03/2015, Duração 1h45m+10min (com consulta)</b>									

Nome: \_\_\_\_\_ Nº Aluno: \_\_\_\_\_

Responda a todas as questões utilizando exclusivamente as folhas agraphadas fornecidas. **Não escreva nada em qualquer outra folha durante a realização do exame.** Suponha que foram realizados as inclusões das bibliotecas necessárias (`#include`) e a instrução: `using namespace std;`. Caso considere que a informação fornecida é insuficiente indique os pressupostos que assumiu na sua folha de teste.

Responda às seguintes questões, preenchendo a tabela com a **opção correcta (em maiúsculas)** (Correcto: x Val / Errado: -x/3 Val).

1	2	3	4	5	6	7	8	9	10	11	12

## GRUPO I (5.0 Valores)

1) Suponha a seguinte classe CPonto:

```
1 class CPonto {
2     private:
3         double x; double y;
4     public:
5         CPonto();
6         CPonto(double x, double y);
7         CPonto(const CPonto& out);
8         ~CPonto();
9         void Transla(double dx, double dy);
10        void Escala(double fx, double fy);
11        double DistAte(const CPonto& out);
12    };
```

Qual das seguintes alternativas é verdadeira:

- A) A linha 2 tem um erro (deveria ser *public* e não *private*).
- B) A linha 3 declara dois membros função da classe CPonto (uma função `double x` e outra `double y`).
- C) As linhas 5 a 7 contêm a declaração dos métodos da classe.
- D) A linha 12 tem um erro (ponto e vírgula no final que nunca se usa depois de uma chaveta)
- E) Nenhuma das anteriores.

2) Utilizando a classe *string* é possível criar uma string, designada por *x* com um conjunto (por exemplo 100) de símbolos especificado (por exemplo símbolos "#")?

- A) Não porque a classe *string* só existe em C
- B) Sim. Fazendo: `string x(100, '#');`
- C) Sim. Fazendo: `string name(8, 'x');`
- D) Sim. Fazendo: `string name(100, "#");`
- E) Nenhuma das Anteriores

3) Supondo o seguinte código indique o que é escrito no écran.

```
1. class CBase {
2.     public:
3.         CBase() {cout << "A";}
4.         ~CBase() {cout << "B";}
5. };
6. CBase x1;
7. int main()
8. {
9.     cout << "B";
10.    CBase x2;
11.    CBase x3;
12.    return 0;
13. }
```

- A) ABAABBB B) BAABBB C) BAAABBB D) ABABB
- E) Nenhuma das Anteriores

4) Supondo seguinte código indique o que é escrito no écran.

```
#include <string>
#include <iostream>
using namespace std;
int main (){
    string str ("Test String");
    for(string::iterator it = str.begin();
        it != str.end()-7; ++it)
        cout << *it;
    return 0;
}
```

- A) String
- B) Test String
- C) Nada pois dá erro na compilação
- D) Test
- E) Nenhuma das anteriores

5) O que escreve no écran o seguinte código?

```
int a = 100, b = 200, *p = &a, *q = &b;
*q = *p; p = q;
cout << *p << *q;
```

- A) 100100 B) 200200 C) 100200 D) 200100
- E) Nenhuma das anteriores

6) Como declarar uma variável *ap* que seja um apontador para um valor constante inteiro?

- A) `int * const ap;`
- B) `int const * const ap;`
- C) `int const *ap;`
- D) `int *ap;`
- E) Nenhuma das anteriores

7) Supondo a seguinte declaração, escolha a opção correcta

```
int* x, y;
```

- A) *y* é um apontador para um inteiro e *x* é um inteiro
- B) *x* e *y* são apontadores para inteiros
- C) *x* e *y* são inteiros
- D) *x* é um apontador para um inteiro e *y* é um inteiro
- E) Nenhuma das anteriores

8) Executando o seguinte código, o que é escrito no écran?

```
void x(int a=10, int b=5){ cout << a << b; }
int main(){
    x(1);
    x(3, 4);
    return 0;
}
```

- A) 105105 B) 10134 C) 1534 D) 1134
- E) Nenhuma das anteriores

9) Considere o seguinte fragmento de código C++ e indique a resposta correta.

```
1 class CRectangle {
2     private:
3         float width, height;
4     public:
5         void set_values (float a, float b) {
6             width=a; height=b; }
7 };
8 class CSquare {
9     private:
10        float side;
11    public:
12        void convert_rect(CRectangle a) {
13            side = (a.width+a.height)/2.0; }
14        friend class CRectangle;
15 };
```

- A) A função `convert_rect` (linhas 12 a 13) não pode usar os membros-dado `width` e `height`, da classe `CRectangle`, (que são privados) por isso está erradamente declarada
- B) O programa está correto e permite converter retângulos em quadrados
- C) O programa não está correto pois falta a declaração `friend class CSquare;` na classe `CSquare`
- D) A função `friend class` (linha 14) não está corretamente declarada
- E) Nenhuma das anteriores

10) Supondo a classe `CAluno`, indique qual das alternativas é verdadeira:

```
1 class CAluno {
2     int numAluno;
3     string nome;
4     double nota;
5     public:
6         CAluno();
7         CAluno(int, string);
8         ~CAluno(int);
9         double getNota() const;
10        int getNumAluno() const;
11        void setNumAluno(int);
12        string getNome() const;
13 };
```

- A) Os dados da classe são públicos.
- B) A linha 8 tem um erro (o destrutor não pode ter parâmetros).
- C) A linha 11 tem um erro (falta o `const` antes do `;` tal como sucede com as linhas 9, 10 e 12).
- D) As linhas 2 a 4 contêm a declaração dos métodos da classe.
- E) Nenhuma das anteriores.

11) Suponha o seguinte fragmento de código, escolha a opção correta:

```
1 class Um {
2     private:
3         int a;
4         double b;
5     public:
6         Um();
7         ~Um();
8         int mudaValor(double p) const {
9             b = p; return a;
10        }
11        double getB() const;
12 };
13 };
```

- A) Os membros-função não podem ser declarados na interface da classe.
- B) A função `mudaValor` altera o valor de `b` e retorna o valor de `a`.
- C) A linha 12 tem um erro (não pode ter `const` antes do `;`).
- D) A linha 8 não deveria ter `const`, para que o código possa ser compilado sem dar erro.
- E) Nenhuma das anteriores.

12) Supondo o seguinte código indique o que é escrito no ecrã.

```
1. class Um {
2.     private:
3.         int a;
4.     public:
5.         Um(int b=2) { a = b; }
6.         void troca(int b) { b = a; }
7. };
8. class Dois {
9.     private:
10.        int a;
11.    public:
12.        Dois(int b=2) { a = b; }
13.        void troca(int &b) { b = a; }
14. };
15. int main()
16. {
17.     Dois c2;
18.     int c = 3;
19.     c2.troca(c);
20.     cout << c;
21.     Um c1(1);
22.     c1.troca(c);
23.     cout << c;
24.     return 0;
25. }
```

- A) 22   B) 21   C) 23   D) 32   E) Nenhuma das Anteriores

## GRUPO II (11.0 Valores)

Considere que se pretende fazer o controlo das contas existentes nas diversas agências de um banco. **Nota:** as classes apresentadas abaixo estão incompletas. Pode adicionar os métodos auxiliares que considerar necessários para a resolução do problema. Suponha que foram incluídos todos os `#include` necessários.

```
class CBanco {
    string nome;
    vector <CAgencia> agencias;
};
```

```
class CAgencia {
    string nomeAgencia;
    string morada;
    vector <CConta> contas;
public:
    CAgencia(string, string);
    int adicionarConta(int, string);
    void listarContas(ostream &os);
};
```

```
class CConta {
    int cod_conta;
    string nome;
    double saldo;
public:
    CConta(int, string);
```

```
    double getSaldo() const;
    string getNomeCliente() const {return nome;}
};
```

2.1) Implemente os construtores das classes **CConta** e **CAgencia**.

2.2) Implemente o método *getSaldo* da classe **CConta**.

2.3) Implemente na classe **CAgencia** o método *adicionarConta*. Este método deve retornar o número de contas existentes na agência, após a adição da nova conta.

2.4) Implemente o método *listarContas* da classe **CAgencia**. Este método deve apresentar em cada linha, o código da conta, nome do titular e respectivo saldo.

2.5) Implemente na classe **CBanco** um novo método: *void imprimeContasNegativas()*. Este método imprime no standard output (monitor) informação sobre as contas negativas existentes em cada agência. Para cada agência deve indicar o seu nome e a informação sobre os titulares das contas (cod\_conta e nome) das contas com saldo inferior a zero.

2.6) Implemente na classe **CConta** o operador - *CConta operator- (int quantia)*. A operação - (que retira uma dada quantia da conta) só pode ser realizada se o saldo da conta não ficar negativo!

2.7) Implemente na classe **CConta** o operador >, ou seja: *bool operator > (const CConta &conta2)*. Este operador permite comparar duas contas, tendo em conta o valor do seu saldo.

2.8) Implemente na classe **CAgencia** o método: *void ordenaContas()*. Este método ordena o vector *contas*, da agência, crescentemente, pelo nome dos titulares das contas.

2.9) Implemente uma função *main()* (programa principal) que lhe permita testar as 8 alíneas anteriores.

### GRUPO III (4.0 Valores)

Pretende-se guardar informação sobre uma colecção de filmes com a seguinte definição:

- Uma colecção é composta por um conjunto de filmes.
- Cada filme possui um título, ano de estreia, realizador, um conjunto de atores e um conjunto de classificações dadas pelos utilizadores do sistema.
- Uma classificação de um utilizador a um filme, contém um valor numérico (1 a 5) e um comentário opcional (texto até 255 caracteres). Contém também indicação de qual o utilizador que a realizou.
- Para cada ator interessa unicamente o seu nome, ano de nascimento e nacionalidade.
- O sistema armazena ainda os dados de cada utilizador que incluem a sua identificação (código numérico), nome, password (código alfanumérico com 8 caracteres) e o número de filmes que já classificou.
- Existem ainda um conjunto de categorias de filmes (“comédia”, “ficção”, “drama”, etc) e cada filme tem uma dada categoria.

Defina um conjunto de classes que na sua opinião melhor descreve o cenário acima, especificando os dados, construtores e destrutores (se necessário) de cada classe. Implemente ainda métodos get e set, exemplificativos, para uma única classe à sua escolha Justifique as suas escolhas. *Nota: Não é necessário implementar qualquer método, simplesmente definir os ficheiros \*.h das classes respetivas.*