

 Universidade do Minho	Escola de Engenharia da Universidade do Minho Mestrado Integrado em Eng. Electrónica Industrial e Computadores Complementos de Programação de Computadores	2014/2015 MIEEIC (1º Ano) 2º Sem
Luís Paulo Reis		
Aula Prática 3a: Exercícios de Introdução à Programação Orientada a Objectos		

Objectivos:

Esta Folha de Exercícios destina-se a:

- Compreender os conceitos de classes e encapsulamento, dados público e dados privados.
- Compreender os conceitos de operador e sobrecarga de operadores.

Os exercícios aqui propostos deverão ser realizados no mais simples ambiente de desenvolvimento possível para a linguagem C: editor de texto de programação ou editor DevC++ e ferramentas da GCC (GNU Compiler Collection) e afins.

Exercício 3a

a1) Escreva a classe **CDate** com a interface indicada (ficheiro CDate.h).

```
// Declaration of the Date class.

#ifndef DATE1_H
#define DATE1_H
class CDate {
public:
    CDate(int = 1, int = 1, int = 1900); // constructor
    ~CDate();
    void print() const; // print date in day/month/year format
private:
    int month; // 1-12
    int day; // 1-31 based on month
    int year; // any year
    int checkDay(); //utility function to test day for month/year
}; // end class CDate

#endif
```

a2) Escreva a implementação da classe **CDate** (ficheiro CDate.cpp) incluindo o construtor, destrutor, a função print e a função checkDay que verifica o número de dias de um dado mês e ano. Relembra-se que um ano é bissexto caso seja múltiplo de 4 e não se que seja múltiplo de 100. No entanto se for múltiplo de 400 é bissexto.

a3) Escreva um programa teste para a classe **CDate** (main_Datas.cpp) que forneça os seguintes resultados:

```
CDate d1(25, 12, 2005), d2(25, 12), d3(25), d4.
d1.print(); // imprime: 25/12/2005
d2.print(); // imprime: 25/12/1900
```

```
d3.print(); // imprime: 25/1/1900
d4.print(); // imprime: 1/1/1900
```

a4) Escreva uma nova função membro pública `setDate(int, int, int)` que receba três inteiros e altere a data para a data respectiva (caso seja válida). Se algum dos valores não for válido deverá ser colocado o valor por defeito (1, 1 ou 1900) no respectivo parâmetro.

a5) Escreva uma nova função membro pública `int difDate(CDate d2)` que calcule a diferença (em dias - inteiro) entre a data e uma outra data recebida como parâmetro. *Nota: Este exercício não é fácil.*

b) Implemente os seguintes operadores para a Classe CDate:

b1) Implemente o operador de comparação "==":

```
bool CDate::operator== (const CDate& rhs);
```

b2) Implemente o operador `ostream& operator<< (ostream& os, CDate &d)` de forma a permitir efectuar:

```
cout << d1; // imprime: 25/12/2005
cout << d2; // imprime: 25/12/1900
cout << d3; // imprime: 25/1/1900
```

b3) Implemente o operador prefixo de incremento "++" para a classe CDate:

```
CDate& CDate::operator++();
```

b4) Implemente o operador de soma de um dado número de dias (inteiro a uma data) para a classe CDate:

```
CDate& CDate::operator+(const int nd);
```

b5) Teste na função main, devidamente, todos os operadores desenvolvidos.