Object-Oriented Programming

Lab 2
ENSIA 2021/2022

Exercise 1 (45 minutes)

1. Modify the following `Date` class code so as to perform error checking on the initializer values for data members `month`, `day` and `year`.
2. Also, provide a member function `nextDay` to increment the day by one. The `Date` object should always remain in a **consistent** state.
3. Write a program that tests function `nextDay` in a loop that prints the date during each iteration to illustrate that `nextDay` works correctly. Be sure to test the following cases:
   a. Incrementing into the next month.
   b. Incrementing into the next year.

Hints:

i. Provide all the `set` and `get` functions.
ii. Provide a function `isLeapYear` which returns `true` if the considered year is a leap year. (Note that a leap year is a multiple of 400, or a multiple of 4 that is not a multiple of 100)
iii. Provide a function `monthDays` which returns the number of days in the considered month.

Header file `lab2_date.h`:

```
#ifndef LAB2_DATE_H
#define LAB2_DATE_H

// Simple Date class
class Date {
    public:
        // Complete the default constructor prototype to initialize
        // the default date to 1/1/1990
        Date( int ???, int ???, int ??? );
        void print();
        // Add the prototypes of the missing functions
        ???
    private:
        int month;
        int day;
        int year;
};

#endif // LAB2_DATE_H
```

Implementation file `lab2_date.cpp`:

```cpp
#include "lab2_date.h"
#include <iostream>
using namespace std;

// Simple Date constructor with no range checking
// You need to modify this constructor to add checks
Date::Date( int m, int d, int y ) {
    month = m;
    day = d;
    year = y;
}

// Print the Date in the form mm-dd-yyyy
void Date::print() {
    cout << month << '-' << day << '-' << year;
}

// Add the implementation of the missing member functions
???
```

Driver file `lab2_main.cpp`:

```cpp
#include "lab2_date.h"
#include <iostream>
using namespace std;

int main() {
    const int MAXDAYS = 160;
    Date date1( 10, 2, 1998 ), date2; // date2 defaults to 1/1/90

    cout << "date1 = ";
    date1.print();
    cout << "\ndate2 = ";
    date2.print();
    cout << endl;

    for ( int loop = 1; loop <= MAXDAYS; ++loop ) {
        // Write call to nextDay
        ???
    }
    return 0;
}
```

Exercise 2 (45 minutes)

Write a program that places a cat into a room, which can hold up to seven cats, each time a key (e.g. Enter) is pressed. Each cat is an object instantiated from class `Cat` - defined in file `lab2_cats.h`. Each cat object, when dynamically instantiated, randomly chooses a *fur*[1] *colour*

[1] fur = fourrure.

(*i.e.*, "black", "grey", or "brown"), an *eye colour* (*i.e.*, "green", "blue", or "brown") and a *hair length* (*i.e.*, "short" or "long"). Provide `get` and `set` methods for these attributes. Carefully consider which `get` and `set` methods should be public and which should be private – all member functions are defined in file `lab2_cats.cpp`.

Certain combinations of cats result in fights breaking out. For example, if the number of grey cats exceeds the number of brown cats, the grey cats fight with the brown cats. A fight also breaks out if at least one black cat with green eyes and at least one black cat with blue eyes are in the room with exactly one black cat with brown eyes. Write a non-member function named check that determines if a fight has broken out.

You are given below a skeleton of code that you should complete.

Header file `lab2_cats.h`:

```
#ifndef LAB2_CATS_H
#define LAB2_CATS_H

// Simple Cat class
class Cat {
    public:
        ???
    private:
        ???
};
#endif
```

Implementation file `lab2_cats.cpp`:

```
#include "lab2_cats.h"
// Constructor
???

// The get functions one after the other
???

// The set functions one after the other. The first statement of the first
   method
// is given; complete all.
??? ???::setFurColour() {
    int x = rand() % 3;
    // Complete it.
    ???
}
```

Driver file `lab2_main.cpp`:

```
// Complete everything that needs to be here.
#include "lab2_cats.h"
#include <iostream>
using namespace std;

int main() {
    // Any variables that need to be declared and possibly initialised
```

```
8       ???
9
10      /* A loop in which you will construct a new cat object, print this new cat
11         object's attributes, and call function check. */
12      for ( int count = 0; count <= numberOfCats; count++ ) {
13          cout << "Press a key to add the next cat to the room." << endl;
14          cin.get();
15
16          ???
17      }
18
19      /* Write a statement to delete all cat objects. */
20      ???
21
22      return 0;
23  }
```