

CSCI 135 Software Design and Analysis I

Homework 5

Due 11/07/08

Saad Mneimneh
Visiting Professor
Hunter College of CUNY

Problem 1: Date class

Define a class called Date to represent the concept of a date. Your class should have three integers as private member data representing the month, the day, and the year. You should also provide the following:

- A constructor that accepts three integers and sets the month, day, and year accordingly.
- A constructor that accepts two integers and sets the month and the day, and sets the year to the current year.
- A default constructor that sets the month, day, and year to the current date.
- A member function that accepts three integers and sets the month, day, and year accordingly (this may be used by all constructors).
- Appropriate member functions to return information about the date, i.e. the month, the day, and the year.

The constructors and the set function must perform appropriate checking. Here's a suggested piece of code that you can use/modify to help you check if a given date is valid:

```
bool leap(int y) {
    //you do it
}

int lastday(int m, int y) {
    int mdays[12]={31,28,31,30,31,30,31,31,30,31,30,31};
    if (leap(y))
        mdays[1]=29; //feb
    return mdays[m-1];
}
```

```

bool valid(int m, int d, int y) {
    if (m<1 || m>12 || y<1 || d<1 || d>lastday(m,y))
        return false;
    else
        return true;
}

```

The following program shows an example of how to get information about the current date (you will need this for the constructors, especially the default constructor).

```

#include <ctime>

int main() {
    time_t t=time(0);
    tm * p=localtime(&t);
    cout<<p->tm_mon+1<<'/'>> //that's the current month
    cout<<p->tm_mday<<'/'>> //that's the current day
    cout<<p->tm_year+1900<<'\'n'>> //that's the current year
}

```

Problem 2: Operator overloading

Provide the following operator overloading functions as member functions of the Date class:

- operator ++ advances the date by one day (pre and post)
- operator -- retracts the date by one day (pre and post)
- operator - takes two dates and returns their difference in number of days

Provide the following operator overloading functions as functions outside the Date class:

- operator + takes a date d and an integer n and returns a date corresponding to date d plus n days
- operator - takes a date d and an integer n and returns a date corresponding to date d minus n days
- operator << takes an output stream and a date and outputs the date in the format m/d/y

The following must work:

```

int main() {
    Date d1=Date(1,13,1973);
    Date d2;
    d2=d1+3;
    d2=3+d1;
    d2=d1-3;
    d2=d1+3;
    cout<<d1<<d2;
}

```

PS: Use pass by reference and const as much as you can.

Problem 3: String reversal

Write a function that reverses a string:

```
void reverse(char * t, int n, const char * s) {...}
```

The string *s* is the original string. The string *t* of length *n* is the one that will hold *s* reversed. Make sure to check that the length of *t* is enough to hold *s* reversed. If not, *t* will hold as many characters as it can fit.

Problem 4: String censoring

Write a function that will censor a string using a dictionary of bad words.

```
void censor(char * s, const char * dict[], int n) {...}
```

In the above function, *dict* is an *n*-element array of strings. Here's an example of how the function can be used:

```
int main() {
    char s[]='fwording shit! what the fword is happening?';
    const char * dict[2]={'shit', 'fword'};
    censor(s,dict,2);
    cout<<s;
}
```

The output should be: *****ing *****! what the ***** is happening?

Problem 5: Polygon

Consider the following simple Point class:

```
class Point {
public:
    float x;
    float y;
}
```

Define a class Polygon that consists of a number of Points.

```
class Polygon {
    Point * points; //the points
    int size; //number of points

public:
    Polygon() {...};
    Polygon(const Point * pts, int n) {...};

    Point getPoint(int n)const {
        //returns the nth point
    }

    void setPoint(const Point& p, int n) {
        //sets the nth point
    }
}
```

```
    int getSize()const {  
        return size;  
    }  
    . . .  
};
```

(a) Complete the class and declare the destructor, the copy constructor, and the assignment operator. Assume that the default constructor constructs an empty polygon.

(b) Change the class Point to have both of its members private, and add a constructor Point(int, int) to construct a point given the two parameters x and y . By doing so, the Polygon class will be affected and will not compile. Discover why and fix the problem.