

Exercice 1 :

Dans le programme suivant la construction d'un objet `CFileImage` est coûteuse puisqu'elle oblige à lire le fichier.

Modifier le programme suivant en utilisant la technique du ... , de façon à ce que la construction coûteuse d'un objet `CFileImage` ne se fasse que lorsque le client appelle une des méthodes `IsValid()` ou `Print()`.

```
#include <memory.h>    // for memset
#include <iostream.h>  // for cout
#define BUF_SIZE 10000

struct CFileImage
{
    CFileImage( const char * fname );
    bool IsValid(){ return m_bValid; }
    void Print();
private:
    bool    m_bValid;
    char    m_buffer[BUF_SIZE];
    size_t  m_buffersize;
};

CFileImage::CFileImage( const char * fname ) :
    m_bValid( false ), m_buffersize( 0 )
{
    memset( m_buffer , 0 , BUF_SIZE );
    FILE * fhandle = fopen( fname , "r" );
    if( fhandle )
    {
        m_bValid = true;
        m_buffersize = fread( m_buffer, sizeof( char ), BUF_SIZE, fhandle );
        fclose( fhandle );
    }
}

void CFileImage::Print()
{
    if( m_bValid )
        cout << m_buffer;
    cout.flush();
}

void main()
{
    CFileImage * pf = new CFileImage("toto.txt");
    pf->Print();
    delete pf;
    // attendre un caractère puis entrée
    char wait;
    cin >> wait;
}
```