

## Déclaration de la classe « complexe »

```
#ifndef COMPLEXE_H
#define COMPLEXE_H

#include <iostream>

using namespace std;
class complexe
{
    double m_r, m_i;

public:
    complexe();
    complexe(double reel, double img);
    complexe(const complexe &comp);
    ~complexe();
    double getReel() const;
    double getImg() const;

    void setReel(double reel);
    void setImg(double img);

    complexe &operator+(const complexe &C2);
    complexe &operator+(double k);
    complexe &operator=(const complexe &C2);
    complexe &operator*(double k);
    complexe &operator*(const complexe &C2);
    string printComplexe();
};

#endif // COMPLEXE_H
```

### A FAIRE

1. Compléter le code ci-dessous
2. Modifier le code de façon à implémenter des opérateurs externes à la classe; en choisir un voire deux.
3. Ajouter l'opérateur interne ou externe permettant d'effectuer une division de nombres complexes (utiliser le complexe conjugué)
4. Proposer une méthode renvoyant le module et l'argument du nombre complexe en écriture exponentielle. On pourra par exemple, ajouter deux attributs *teta* pour l'argument et *ro* pour le module et créer deux accesseurs en lecture (get)

## Implémentation des méthodes de la classe "complexe"

```
#include "complexe.h"

complexe::complexe()
{
}

complexe::complexe(double reel, double img) : m_r(    ), m_i(    )
{
}

complexe::complexe(const complexe &comp)
{
    m_r =    ;
    m_i =    ;
}

complexe::~~complexe()
{
}

double complexe::getReel() const
{
    return m_r;
}

double complexe::getImg() const
{
    return m_i;
}

complexe &complexe::operator+(const complexe &C2)
{
    complexe *C;
    C = new complexe;
    C->setReel(    );
    C->setImg(    );

    return *C;
}

complexe& complexe::operator+(double k)
{
    complexe *C;
    C = new complexe;
    C->setReel(    );
    C->setImg(    );
    return *C;
}

complexe &complexe::operator=(const complexe &C2)
{
    m_r =    ;
    m_i =    ;
    return *this;
}
```

```

complexe &complexe::operator*(double k)
{
    complexe *C;
    C = new complexe;
    C->setReel( );
    C->setImg( );
    return *C;
}

complexe &complexe::operator*(const complexe &C2)
{
    //(a+ib)(c+id) = ac-bd + i(ad+bc)

    return *this;
}

void complexe::setReel(double reel)
{
    m_r = reel;
}

void complexe::setImg(double img)
{
    m_i = img;
}

string complexe::printComplexe()
{
    string ch;
    //std::string mch;
    if (m_i < 0)
    {
        ch = std::to_string(m_r); //New in C++11
        ch += "-";
        ch += std::to_string (-1*m_i);
        ch += "i";
    }
    else
    {
        ch = std::to_string(m_r);
        ch += "+";
        ch += std::to_string (m_i);
        ch += "i";
    }

    return ch;
}

```