

COMPTE RENDU DE CHAPITRE II + TPS



Réalisé par :
Abderrahmane ETTOUNANI
II-BDCC

Encadré par :
Monsieur K.MANSOURI

les exemples de cours & tps

-----compteur-----

```
#include <iostream>
#include <stdlib.h>
#include <math.h>
using namespace std;
You, 15 seconds ago | 1 author (You)
class Compteur{
    static int ctr;
public:
    Compteur();
    ~Compteur();
};
int Compteur ::ctr = 0;
Compteur ::Compteur(){
    cout << " un nouvel objet vient de sa créer " << endl;
    cout << " il y a maintenant : " << ++ctr << " objets ";
}
Compteur ::~Compteur(){
    cout << " un objet vient de se détruire " << endl;
    cout << " il reste maintenant  : " << --ctr << " objets ";
}
void essai(){
    Compteur u, v;
}
int main(){
    essai();
    Compteur a;
    essai();
    Compteur b;
    return 0;
}
You, 1 hour ago • add compteur ...
```

```
un nouvel objet vient de sa créer
il y a maintenant : 1 objets  un nouvel objet vient de sa créer
il y a maintenant : 2 objets  un objet vient de se détruire
il reste maintenant  : 1 objets  un objet vient de se détruire
il reste maintenant  : 0 objets  un nouvel objet vient de sa créer
il y a maintenant : 1 objets  un nouvel objet vient de sa créer
il y a maintenant : 2 objets  un nouvel objet vient de sa créer
il y a maintenant : 3 objets  un objet vient de se détruire
il reste maintenant  : 2 objets  un objet vient de se détruire
il reste maintenant  : 1 objets  un nouvel objet vient de sa créer
il y a maintenant : 2 objets  un objet vient de se détruire
il reste maintenant  : 1 objets  un objet vient de se détruire
il reste maintenant  : 0 objets
```

-----exercice 1-----

```
class Point{
    int x, y, couleur;
public:
    void initialiser(int, int, int);
    void deplacer(int, int);
    void afficher();
    void effacer();
};


void Point::initialiser(int abs, int ord, int c){
    x = abs;
    y = ord;
    couleur = c;
}

void Point::deplacer(int dx, int dy){
    effacer();
    x = x + dx;
    y = y + dy;
    afficher();
}

void Point::afficher(){
    cout << "je suis en " << x << " " << y << endl;
}

void Point::effacer(){
    int aux = couleur;
    couleur = 1;
    afficher();
    couleur = aux;
}

int main(){
    Point p1;
    p1.initialiser(10, 30, 1);
    p1.afficher();
    p1.deplacer(20, 40);
    p1.effacer();
    return 0;
}
```



```
je suis en 10 30
je suis en 10 30
je suis en 30 70
je suis en 30 70
```

-----exercice 2-----

```
class Point{
    int x, y, couleur;
public:
    void initialiser(int, int, int);
    void deplacer(int, int);
    void afficher();
    void effacer();
};

void Point::initialiser(int abs, int ord, int c){
    x = abs;
    y = ord;
    couleur = c;
}

void Point::deplacer(int dx, int dy){
    effacer();
    x = x + dx;
    y = y + dy;
    afficher();
}

void Point::afficher(){
    cout << "je suis en " << x << " " << y << endl;
}

void Point::effacer(){
    int aux = couleur;
    couleur = 2; // black
    afficher();
    couleur = 1;
}

void scene(){
    Point u;
    u.initialiser(10, 30, 1);
    u.afficher();
    u.deplacer(20, 40);
    u.afficher();
    u.effacer();
}

int main(){
    scene();
    return 0;}

```




```
je suis en 10 30
je suis en 10 30
je suis en 30 70
je suis en 30 70
```



-----exercice 3-----

```
class Point{
    int x, y, couleur;
public:
    Point();
    void deplacer(int, int);
    void afficher();
    void effacer();
};
Point::Point(){
    x = 0;
    y = 0;
    couleur = 0;
}
void Point::deplacer(int dx, int dy){
    effacer();
    x = x + dx;
    y = y + dy;
    afficher();
}
void Point::afficher(){
    cout << "je suis en " << x << " " << y << endl;
}
void Point::effacer(){
    int aux = couleur;
    couleur = 2; // black
    afficher();
    couleur = 1;
}
void scene(){
    Point u;
    u.afficher();
    u.deplacer(20, 40);
    u.effacer();
}
int main(){
    scene();
    return 0;
}
```



```
je suis en 0 0
je suis en 0 0
je suis en 20 40
je suis en 20 40
```

-----exercice 4-----

```
class Point{
    int x, y, couleur;

public:
    Point(int, int, int);
    void deplacer(int, int);
    void afficher();
    void effacer();
};

Point::Point(int abs, int ord, int c){
    x = abs;
    y = ord;
    couleur = c;
}


void Point::deplacer(int dx, int dy){
    effacer();
    x = x + dx;
    y = y + dy;
    afficher();
}

void Point::afficher(){
    cout << "je suis en " << x << " " << y << endl;
}

void Point::effacer(){
    int aux = couleur;
    couleur = 2; // black
    afficher();
    couleur = 1;
}

void scene(){
    Point u(10, 30, 1);
    u.afficher();
    u.deplacer(20, 40);
    u.effacer();
}


int main(){
    scene();
    return 0;
}
```



```
je suis en 47 3
je suis en 47 3
je suis en 67 43
je suis en 67 43
```

-----exercice 5-----

```
class Point{
    int x, y, couleur;
public:
    Point();
    void deplacer(int, int);
    void afficher();
    void effacer();
    ~Point();
};
Point::Point(){
    x = 0;
    y = 0;
    couleur = 0;
}
void Point::deplacer(int dx, int dy){
    effacer();
    x = x + dx;
    y = y + dy;
    afficher();
}
void Point::afficher(){
    cout << "je suis en " << x << " " << y << endl;
}
void Point::effacer(){
    int aux = couleur;
    couleur = 2; // black
    afficher();
    couleur = 1;
}
Point::~~Point(){
    afficher();
}
void scene(){
    Point u;
    u.afficher();
    u.deplacer(20, 40);
    u.effacer();
}
```



```
je suis en 0 0
je suis en 0 0
je suis en 20 40
je suis en 20 40
je suis en 20 40
```

-----exercice 6-----

```
// exercice 6      You, 2 hours ago *
#include <iostream>
#include <stdlib.h>
using namespace std;

You, 2 hours ago | 1 author (You)
class SuitAr
{
    int nbval, *val;

public:
    SuitAr(int, int);
    ~SuitAr();
    void afficher();
};

SuitAr::SuitAr(int nb, int mul)
{
    int i;
    nbval = nb;
    val = new int[nbval];
}

SuitAr::~~SuitAr()
{
    cout << "detruire object"<<endl;
}

int main()
{
    SuitAr ar(2, 4);

    return 0;
}
```

detruire object



-----exercice 7-----

```
#include <iostream>
#include <stdlib.h>
using namespace std;
You, 5 seconds ago | 1 author (You)
class Hasard{
    int nbval;
    int *val;
public:
    Hasard(int, int);
    ~Hasard();
    void affiche();
};
Hasard::Hasard(int nb, int max){
    int i;
    val = new int[nbval = nb];
    for (i = 0; i < nb; i++)
        val[i] = double(rand()) / RAND_MAX * max;
}
Hasard::~~Hasard(){
    delete val;
}
void Hasard::affiche(){
    int i;
    for (i = 0; i < nbval; i++)
        cout << val[i] << " ";
    cout << "\n";
}
int main(){
    Hasard suite1(15, 5);
    suite1.affiche();
    Hasard suite2(6, 12);
    suite2.affiche();
    return 0;
}
```



```
4 1 3 3 4 0 1 3 1 2 2 3 1 2 4
10 7 8 1 7 0
```


-----exercice 8-----

```
#include <iostream>
#include <stdlib.h>
#include <math.h>
using namespace std;
You, 11 seconds ago | 1 author (You)
class Complexe{
private:
    double x, y;
public:
    Complexe();
    Complexe(double, double);
    Complexe(double a);
    void set(double, double);
    double module();
    void affiche();
    double real() const;
    double img() const;
};
void Complexe::set(double a, double b){
    x = a;
    y = b;
}
double Complexe::module(){
    double res;
    res = sqrt(x * x + y * y);
    return res;
}
void Complexe::affiche(){
    printf("%f + %f i\n", x, y);
}
Complexe::Complexe(double a, double b){
    x = a;
    y = b;
}
```

```
Complexe::Complexe(double a){
    x = a;
    y = 0;
}
Complexe::Complexe()
{
    x = 0;
    y = 0;
}
int main()
{
    Complexe c1(15.4, 5.2);
    c1.affiche();

    Complexe c2(5.2);
    c2.affiche();

    double res = c1.module();
    cout << "module : " << res << endl;
    return 0;
}
```



```
15.400000 + 5.200000 i
5.200000 + 0.000000 i
module : 16.2542
```