

Travaux Pratiques

COURS



🚦 Ce TP réalisé par :

ZRAIDI NAJWA
(GLSID1)

Surdéfinition des fonctions membres :

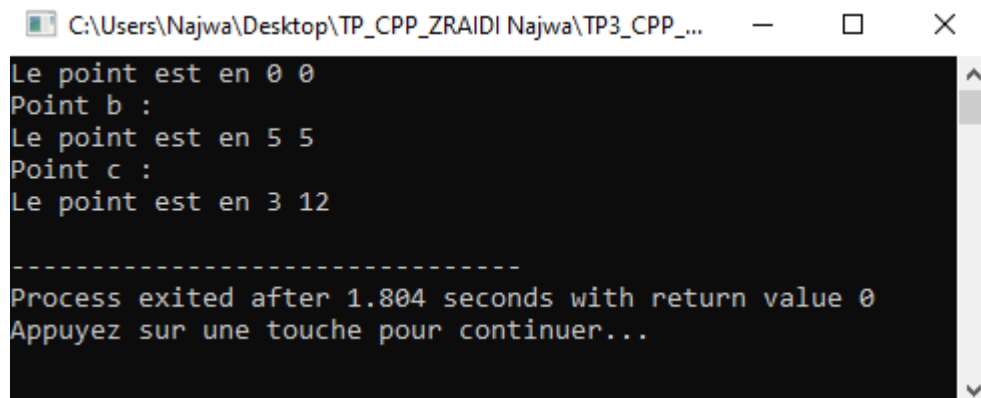
Exemple :

- ✓ Définition de plusieurs constructeurs.

```
//Definition de plusieurs constructeurs
#include<iostream>
#include<conio.h>
using namespace std;
class point {
    int x,y;
    public:
        point();//constructeur 1
        point(int);//constructeur 2
        point(int,int);//constructeur 3
        void affiche();
        void affiche(char *);
};
point :: point() //constructeur 1
{
    x=0;
    y=0;
}
point :: point(int abs) //constructeur 2
{
    x=abs;
    y=abs;
}
point :: point(int abs,int ord) //constructeur 3
{
    x=abs;
    y=ord;
}
void point :: affiche(){
    cout<<"Le point est en "<< x<<" "<<y<<endl;
};
void point :: affiche(char *message){
    cout<<message;
    affiche();
}
int main(){
    system("cls");
    point a;
    a.affiche();
    point b(5);
    b.affiche("Point b : \n");
    point c(3,12);
```

```
c.affiche("Point c : \n");  
getch();  
}
```

✓ L'exécution de programme donne :



```
C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP3_CPP_...  
Le point est en 0 0  
Point b :  
Le point est en 5 5  
Point c :  
Le point est en 3 12  
  
-----  
Process exited after 1.804 seconds with return value 0  
Appuyez sur une touche pour continuer...
```

Fonction membres en ligne :

Exemple :

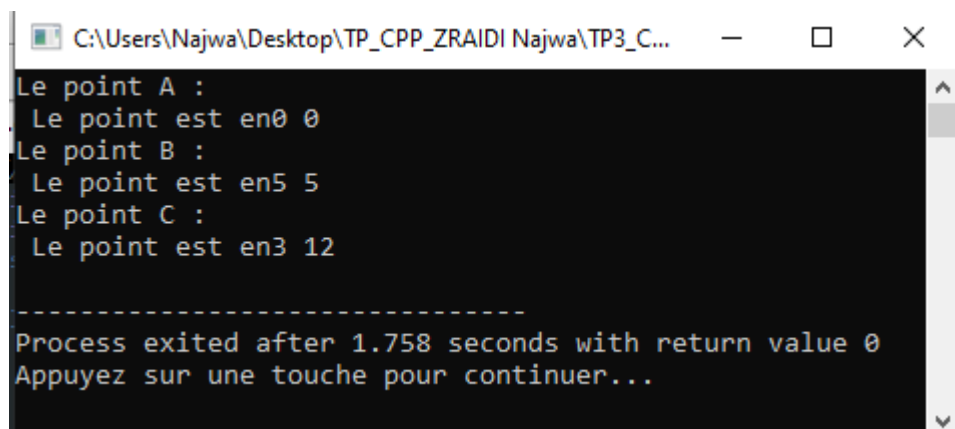
```
//Fonctions Membres en ligne  
#include<iostream>  
#include<conio.h>  
using namespace std;  
  
class point  
{  
    int x,y;  
    public :  
        point(){ //constructeur 1  
            x=0;  
            y=0;  
        }  
        point(int abs) //constructeur 2  
        {  
            x=abs;  
            y=abs;  
        }  
        point(int abs,int ord) //constructeur 3  
        {  
            x=abs;  
            y=ord;  
        }  
        void affiche();  
};  
void point :: affiche(){
```

```

    cout<<"Le point est en"<< x<<" "<<y<<endl;
};
int main(){
    point a,b(5);
    cout<<"Le point A :\n ";
    a.affiche();
    cout<<"Le point B :\n ";
    b.affiche();
    point c(3,12);
    cout<<"Le point C :\n ";
    c.affiche();
    getch(); }

```

✓ L'exécution de programme donne :



```

C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP3_C...
Le point A :
Le point est en0 0
Le point B :
Le point est en5 5
Le point C :
Le point est en3 12

-----
Process exited after 1.758 seconds with return value 0
Appuyez sur une touche pour continuer...

```

⇒ La comparaison entre les méthodes en ligne et les méthodes offline

Exemple test : Méthodes offline

```

//Fonctions off-ligne
#include<iostream>
#include<conio.h>
using namespace std;
class point {
    int x,y;
    public:
        point();//constructeur 1
        point(int);//constructeur 2
        point(int,int);//constructeur 3
        void affiche();
        void affiche(char *);
};

point :: point() //constructeur 1
{
    x=0;
    y=0;
}

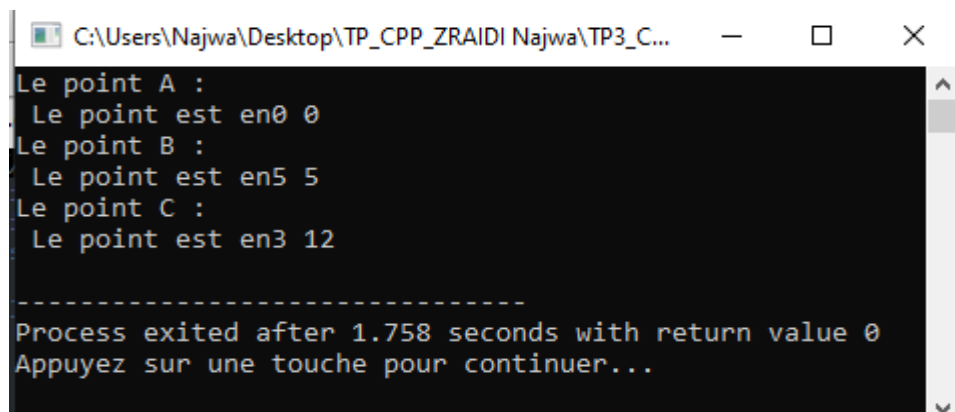
```

```

}
point :: point(int abs) //constructeur 2
{
    x=abs;
    y=abs;
}
point :: point(int abs,int ord) //constructeur 3
{
    x=abs;
    y=ord;
}
void point :: affiche(){
    cout<<"Le point est en "<< x<<" "<<y<<endl;
};
void point :: affiche(char *message){
    cout<<message;
    affiche();
}
int main(){
    point a,b(5);
    cout<<"Le point A :\n ";
    a.affiche();
    cout<<"Le point B :\n ";
    b.affiche();
    point c(3,12);
    cout<<"Le point C :\n ";
    c.affiche();
    getch();
}

```

✓ L'exécution de programme donne :



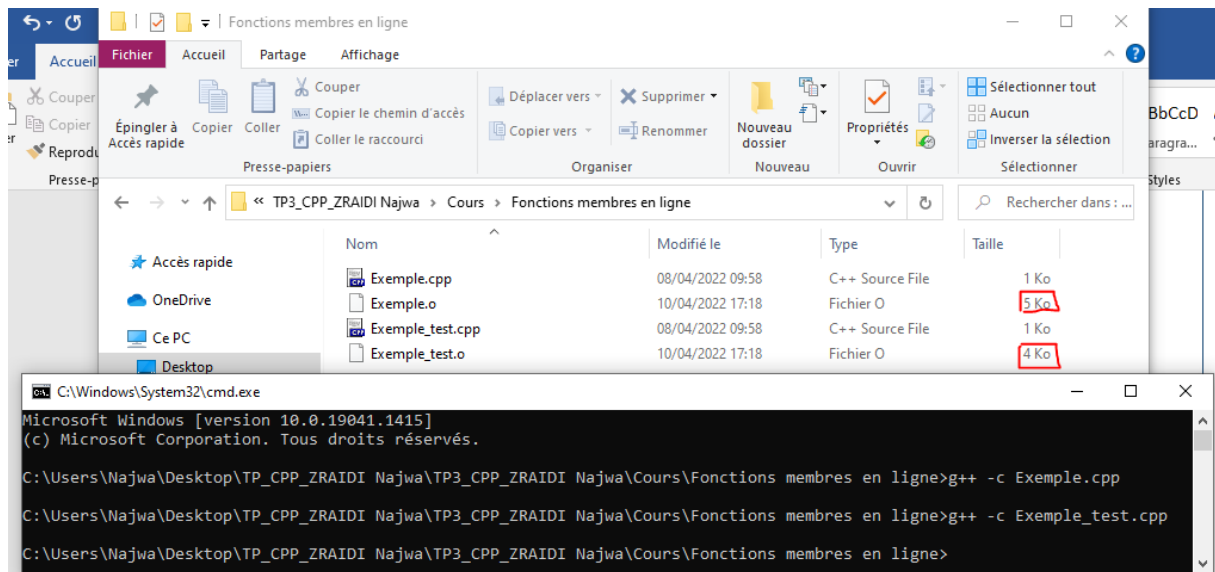
```

C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP3_C...
Le point A :
Le point est en 0 0
Le point B :
Le point est en 5 5
Le point C :
Le point est en 3 12

-----
Process exited after 1.758 seconds with return value 0
Appuyez sur une touche pour continuer...

```

- Le fichier d'objet de exemple qui contient les méthodes en ligne > Le fichier d'objet de exemple qui contient les méthodes offline.



Initialisation des paramètres par défaut :

Exemple :

```

//Fonctions Membres en ligne
#include<iostream>
#include<conio.h>
using namespace std;

class point
{
    int x,y;
    public :
    point(int abs=0,int ord=0) //constructeur
    {
        x=abs;
        y=ord;
    }
    void affiche(char* ="Position du point\n");//argument par défaut
};

void point :: affiche(char *message)
{
    cout<<"Le point est en "<< x<<" "<<y-1<<endl;
    cout<<message;
    cout<<"Le point est en "<< x<<" "<<y<<endl;
}

int main(){
    point a,b(40);
    a.affiche();
}

```

```

b.affiche("Point B\n");
char text[10]="Bonjour \n";
point c(3,12);
c.affiche(text);
getch();
}

```

✓ L'exécution de programme donne :

```

C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP3_...
Le point est en 0 -1
Position du point
Le point est en 0 0
Le point est en 40 -1
Point B
Le point est en 40 0
Le point est en 3 11
Bonjour
Le point est en 3 12

-----
Process exited after 1.608 seconds with return value 0
Appuyez sur une touche pour continuer...

```

Objets transmis en argument d'une fonction membre :

Exemple :

```

//Fonctions Membres en ligne
#include<iostream>
#include<conio.h>
using namespace std;

class point
{
    int x,y;
    public :
    point(int abs=0,int ord=0) //constructeur
    {
        x=abs;
        y=ord;
    }
    void affiche(char* ="Position du point\n");//argument par default
};

void point :: affiche(char *message)
{

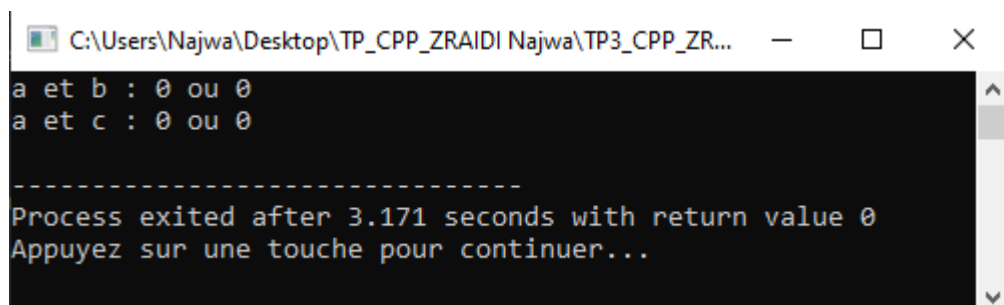
```

```

    cout<<"Le point est en "<< x<<" "<<y-1<<endl;
    cout<<message;
    cout<<"Le point est en "<< x<<" "<<y<<endl;
}
int main(){
    point a,b(40);
    a.affiche();
    b.affiche("Point B\n");
    char text[10]="Bonjour \n";
    point c(3,12);
    c.affiche(text);
    getch();
}

```

✓ L'exécution de programme donne :



```

C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP3_CPP_ZR...
a et b : 0 ou 0
a et c : 0 ou 0

-----
Process exited after 3.171 seconds with return value 0
Appuyez sur une touche pour continuer...

```

Exercice 1 :

```

#include<iostream>
#include<conio.h>
using namespace std;
class point
{
    int x,y;
    public :
    point(int abs=0,int ord=0) //constructeur
    {
        x=abs;
        y=ord;
    }
    int coincide(point *); //passage par adresse
};

int point :: coincide(point *pt){

    if((pt->x==x)&&(pt->y==y))
        /*pt.x acces d'absice de point

```



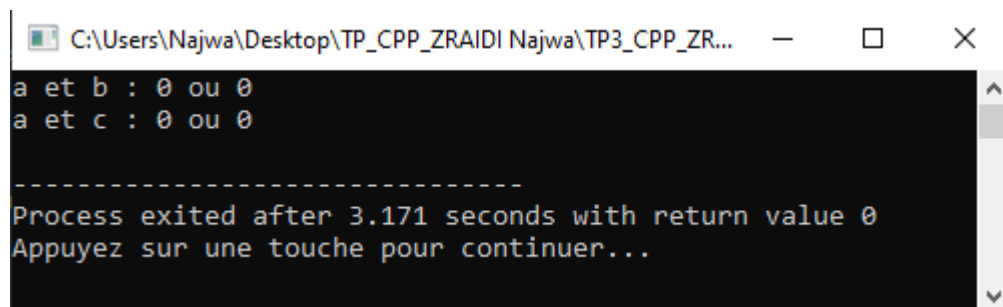
```

    pt et pt.x acces d'ordonnee de point pt*/
    return 1;
else
    return 0;
}

int main(){
    int test1,test2;
    point a,b(1),c(0,2);
    test1=a.coincide(&b);
    test2=b.coincide(&a);
    cout<<"a et b : "<<test1 <<" ou "<<test2<<endl;
    test1=a.coincide(&c);
    test2=c.coincide(&a);
    cout<<"a et c : "<<test1 <<" ou "<<test2<<endl;
    getch();
}

```

✓ L'exécution de programme donne :



```

C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP3_CPP_ZR...
a et b : 0 ou 0
a et c : 0 ou 0

-----
Process exited after 3.171 seconds with return value 0
Appuyez sur une touche pour continuer...

```

Exercice 2 :

```

#include<iostream>
#include<conio.h>
using namespace std;
class point
{
    int x,y;
    public :
    point(int abs=0,int ord=0) //constructeur
    {
        x=abs;
        y=ord;
    }
    int coincide(point &); //passage par reference
};

int point :: coincide(point &pt){

```

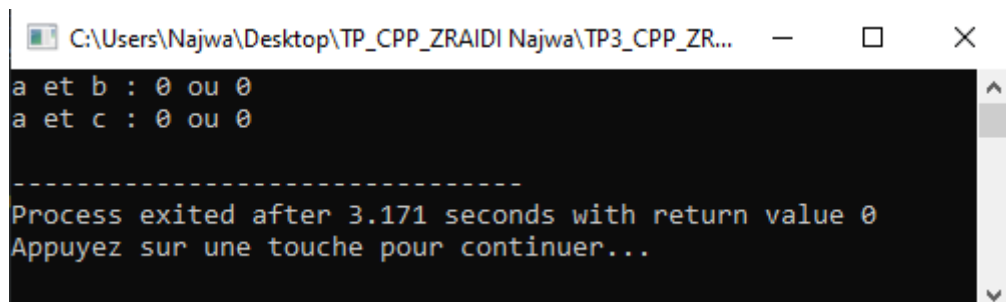
```

    if((pt.x==x)&&(pt.y==y))
    /*pt.x acces d'absce de point
    pt et pt.x acces d'ordonnee de point pt*/
    return 1;
    else
    return 0;
}

int main(){
    int test1,test2;
    point a,b(1),c(0,2);
    test1=a.coincide(b);
    test2=b.coincide(a);
    cout<<"a et b : "<<test1 <<" ou "<<test2<<endl;
    test1=a.coincide(c);
    test2=c.coincide(a);
    cout<<"a et c : "<<test1 <<" ou "<<test2<<endl;
    getch();
}

```

✓ L'exécution de programme donne :



```

C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP3_CPP_ZR...
a et b : 0 ou 0
a et c : 0 ou 0

-----
Process exited after 3.171 seconds with return value 0
Appuyez sur une touche pour continuer...

```

Exercice 3-1 :

```

#include<iostream>
#include<conio.h>
using namespace std;

class vecteur
{
    float x,y;
    public :
        vecteur(float,float);
        void homotethie(float);
        float det(vecteur);
        void affiche();
};

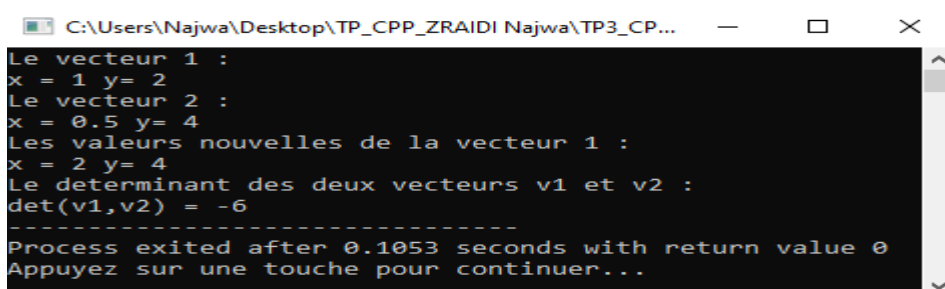
```

```

vecteur :: vecteur(float abs=0.,float ord =0.){
    x=abs;
    y=ord;
}
void vecteur :: homotethie(float val){
    x=x*val;
    y=y*val;
}
void vecteur :: affiche(){
    cout<<"x = "<<x<<" y= "<<y<<endl;
}
float vecteur:: det(vecteur v1)
{
    return v1.x * y - x *v1.y;
}
int main(){
    vecteur v1(1,2),v2(0.5,4);
    cout<<"Le vecteur 1 : "<<endl;
    v1.affiche();
    cout<<"Le vecteur 2 : "<<endl;
    v2.affiche();
    cout<<"Les valeurs nouvelles de la vecteur 1 : "<<endl;
    v1.homotethie(2);
    v1.affiche();
    cout<< "Le determinant des deux vecteurs v1 et v2 : \n";
    cout <<"det(v1,v2) = "<<v1.det(v2);
}

```

✓ L'exécution de programme donne :



```

C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP3_CP...
Le vecteur 1 :
x = 1 y= 2
Le vecteur 2 :
x = 0.5 y= 4
Les valeurs nouvelles de la vecteur 1 :
x = 2 y= 4
Le determinant des deux vecteurs v1 et v2 :
det(v1,v2) = -6
-----
Process exited after 0.1053 seconds with return value 0
Appuyez sur une touche pour continuer...

```

Exercice 3-2 :

```

#include<iostream>
#include<conio.h>
using namespace std;

class vecteur
{
    float x,y;
    public :

```

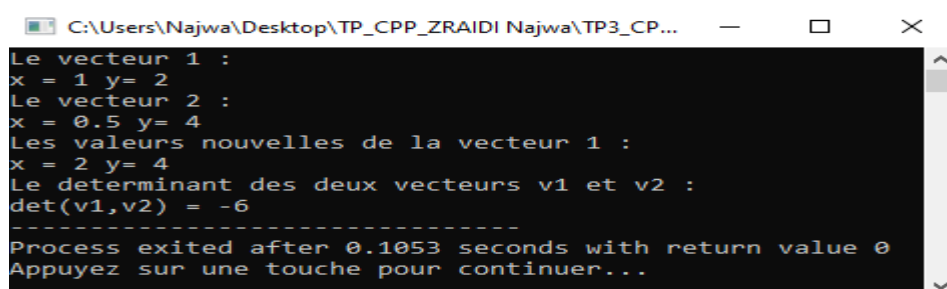
```

    vecteur(float,float);
    void homotethie(float);
    float det(vecteur &);//passage par reference
    void affiche();
};

vecteur :: vecteur(float abs=0.,float ord =0.){
    x=abs;
    y=ord;
}
void vecteur :: homotethie(float val){
    x=x*val;
    y=y*val;
}
void vecteur :: affiche(){
    cout<<"x = "<<x<<" y= "<<y<<endl;
}
float vecteur:: det(vecteur &v1)
{
    return v1.x * y - x *v1.y;
}
int main(){
    vecteur v1(1,2),v2(0.5,4);
    cout<<"Le vecteur 1 : "<<endl;
    v1.affiche();
    cout<<"Le vecteur 2 : "<<endl;
    v2.affiche();
    cout<<"Les valeurs nouvelles de la vecteur 1 : "<<endl;
    v1.homotethie(2);
    v1.affiche();
    cout<<"Le determinant des deux vecteurs v1 et v2 : \n";
    cout <<"det(v1,v2) = "<<v1.det(v2);
}

```

✓ L'exécution de programme donne :



```

C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP3_CP...
Le vecteur 1 :
x = 1 y= 2
Le vecteur 2 :
x = 0.5 y= 4
Les valeurs nouvelles de la vecteur 1 :
x = 2 y= 4
Le determinant des deux vecteurs v1 et v2 :
det(v1,v2) = -6
-----
Process exited after 0.1053 seconds with return value 0
Appuyez sur une touche pour continuer...

```

Exercice 3-3 :

```
#include<iostream>
#include<conio.h>
using namespace std;

class vecteur
{
    float x,y;
    public :
        vecteur(float,float);
        void homotethie(float);
        float det(vecteur *); //passage par reference
        void affiche();
};

vecteur :: vecteur(float abs=0.,float ord =0.){
    x=abs;
    y=ord;
}

void vecteur :: homotethie(float val){
    x=x*val;
    y=y*val;
}

void vecteur :: affiche(){
    cout<<"x = "<<x<<" y= "<<y<<endl;
}

float vecteur:: det(vecteur *v1)
{
    return v1->x * y - x *v1->y;
}

int main(){
    vecteur v1(1,2),v2(0.5,4);
    cout<<"Le vecteur 1 : "<<endl;
    v1.affiche();
    cout<<"Le vecteur 2 : "<<endl;
    v2.affiche();
    cout<<"Les valeurs nouvelles de la vecteur 1 : "<<endl;
    v1.homotethie(2);
    v1.affiche();
    cout<<"Le determinant des deux vecteurs v1 et v2 : \n";
    cout <<"det(v1,v2) = "<<v1.det(&v2);
}
```

✓ L'exécution de programme donne :

```
C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP3_CP...
Le vecteur 1 :
x = 1 y= 2
Le vecteur 2 :
x = 0.5 y= 4
Les valeurs nouvelles de la vecteur 1 :
x = 2 y= 4
Le determinant des deux vecteurs v1 et v2 :
det(v1,v2) = -6
-----
Process exited after 0.1053 seconds with return value 0
Appuyez sur une touche pour continuer...
```

Objets retourne par une fonction membre :

▪ Retour par valeur:

Exemple :

```
#include<iostream>
#include<conio.h>
using namespace std;
// La valeur de retour d'une fonction est un objet
//Transmission par valeur

class point
{
    int x,y;
    public :
    point(int abs=0,int ord=0) //constructeur
    {
        x=abs;
        y=ord;
    }
    point symetrique();
    void affiche();
};

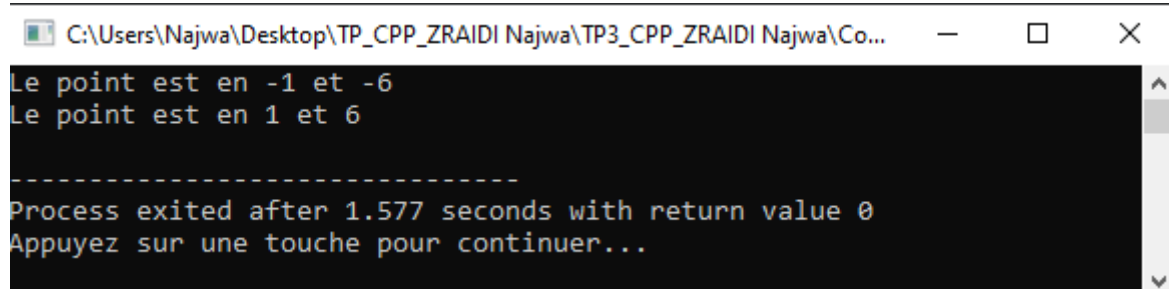
point point :: symetrique(){
    point res;
    res.x=-x;
    res.y=-y;
    return res;
}

void point :: affiche(){
    cout<<"Le point est en "<<x<<" et "<<y<<"\n";
}

int main(){
    point a,b(1,6);
    a=b.symetrique();
    a.affiche();
    b.affiche();
}
```

```
    getch();  
}
```

✓ L'exécution de programme donne :



```
C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP3_CPP_ZRAIDI Najwa\Co...  
Le point est en -1 et -6  
Le point est en 1 et 6  
-----  
Process exited after 1.577 seconds with return value 0  
Appuyez sur une touche pour continuer...
```

▪ Retour par adresse:

Exemple :

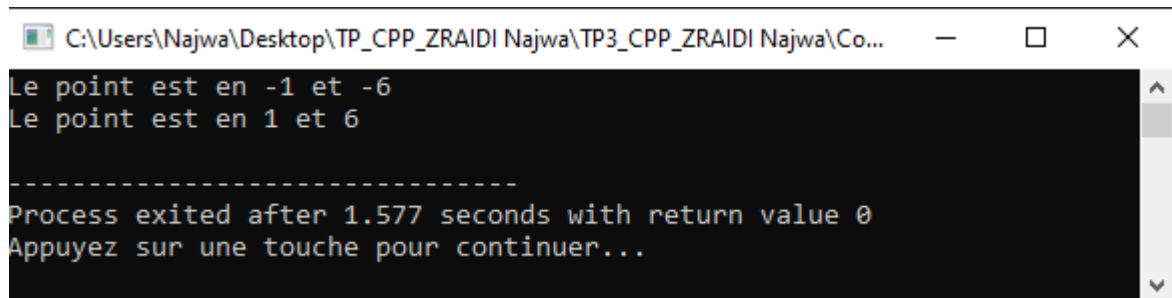
```
#include<iostream>  
#include<conio.h>  
using namespace std;  
// La valeur de retour d'une fonction est un objet  
//Transmission par adresse  
  
class point  
{  
    int x,y;  
    public :  
    point(int abs=0,int ord=0) //constructeur  
    {  
        x=abs;  
        y=ord;  
    }  
    point *symetrique();  
    void affiche();  
};  
  
point *point :: symetrique(){  
    point *res =new point;  
    res->x=-x;  
    res->y=-y;  
    return res;  
}  
  
void point :: affiche(){  
    cout<<"Le point est en "<<x<<" et "<<y<<"\n";  
}  
  
int main(){  
    point a,b(1,6);
```

```

    a=*b.symetrique();
    a.affiche();
    b.affiche();
    getch();
}

```

✓ L'exécution de programme donne :



```

C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP3_CPP_ZRAIDI Najwa\Co...
Le point est en -1 et -6
Le point est en 1 et 6
-----
Process exited after 1.577 seconds with return value 0
Appuyez sur une touche pour continuer...

```

▪ Retour par reference:

Exemple :

```

#include<iostream>
#include<conio.h>
using namespace std;
// La valeur de retour d'une fonction est un objet
//Transmission par reference

class point
{
    int x,y;
    public :
    point(int abs=0,int ord=0) //constructeur
    {
        x=abs;
        y=ord;
    }
    point &symetrique();
    void affiche();
};

point &point :: symetrique(){
    static point res;
    // La variable res est obligatoirement static pour passer par reference
    res.x=-x;
    res.y=-y;
    return res;
}

void point :: affiche(){
    cout<<"Le point est en "<<x<<" et "<<y<<"\n";
}

```

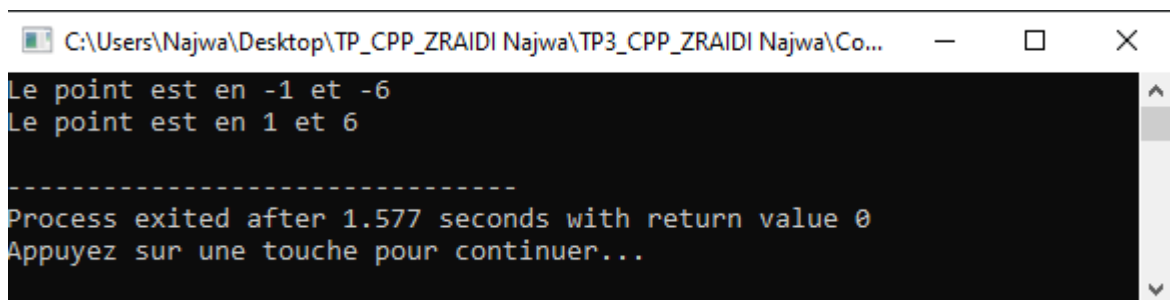


```

}
int main(){
    point a,b(1,6);
    a=b.symetrique();
    a.affiche();
    b.affiche();
    getch();
}

```

✓ L'exécution de programme donne :



```

C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP3_CPP_ZRAIDI Najwa\Co...
Le point est en -1 et -6
Le point est en 1 et 6
-----
Process exited after 1.577 seconds with return value 0
Appuyez sur une touche pour continuer...

```

Exercice a:

```

#include<iostream>
#include<conio.h>
using namespace std;

class vecteur
{
    float x,y;
    public :
        vecteur(float,float);
        vecteur homotethie(float);
        float det(vecteur);
        void affiche();
};

vecteur :: vecteur(float abs=0.,float ord =0.){
    x=abs;
    y=ord;
}

vecteur vecteur :: homotethie(float val){
    x=x*val;
    y=y*val;
}

void vecteur :: affiche(){
    cout<<"x = "<<x<<" y= "<<y<<endl;
}

float vecteur:: det(vecteur v1)

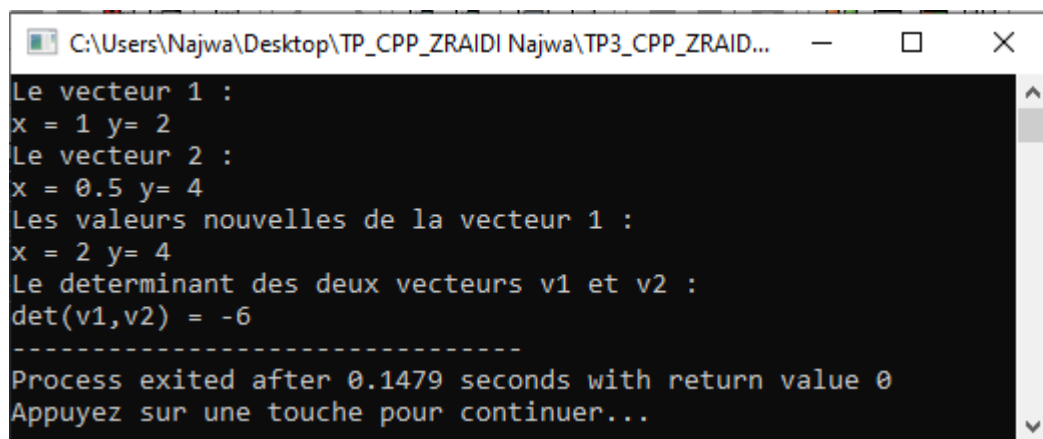
```

```

{
    return v1.x * y - x * v1.y;
}
int main(){
    vecteur v1(1,2),v2(0.5,4);
    cout<<"Le vecteur 1 : "<<endl;
    v1.affiche();
    cout<<"Le vecteur 2 : "<<endl;
    v2.affiche();
    cout<<"Les valeurs nouvelles de la vecteur 1 : "<<endl;
    v1.homotethie(2);
    v1.affiche();
    cout<< "Le determinant des deux vecteurs v1 et v2 : \n";
    cout <<"det(v1,v2) = "<<v1.det(v2);
}

```

✓ L'exécution de programme donne :



```

C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP3_CPP_ZRAID...
Le vecteur 1 :
x = 1 y= 2
Le vecteur 2 :
x = 0.5 y= 4
Les valeurs nouvelles de la vecteur 1 :
x = 2 y= 4
Le determinant des deux vecteurs v1 et v2 :
det(v1,v2) = -6
-----
Process exited after 0.1479 seconds with return value 0
Appuyez sur une touche pour continuer...

```

Exercice b:

```

#include<iostream>
#include<conio.h>
using namespace std;

class vecteur
{
    float x,y;
    public :
        vecteur(float,float);
        vecteur *homotethie(float);
        float det(vecteur);
        void affiche();
};

vecteur :: vecteur(float abs=0.,float ord =0.){
    x=abs;

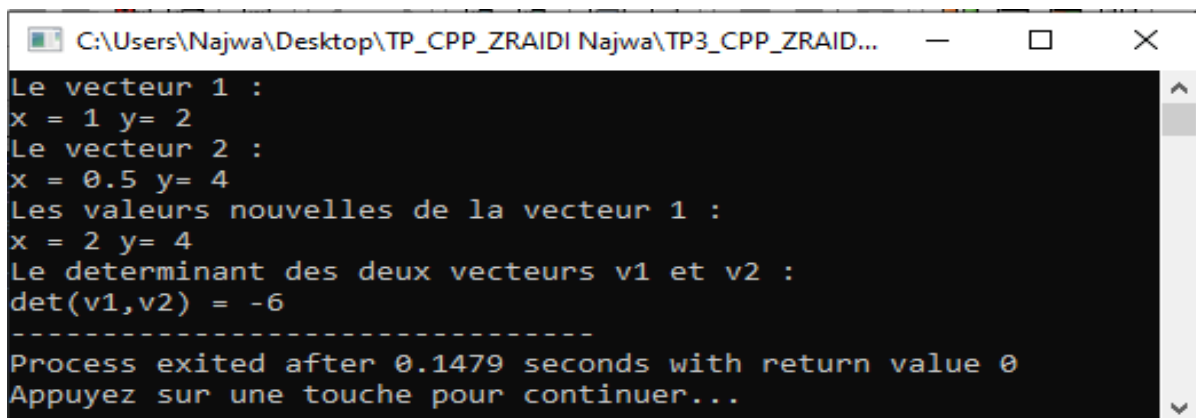
```

```

        y=ord;
    }
    vecteur *vecteur :: homotethie(float val){
        vecteur *res=new vecteur;
        res->x=x*val;
        res->y=y*val;
        return res;
    }
    void vecteur :: affiche(){
        cout<<"x = "<<x<<" y= "<<y<<endl;
    }
    float vecteur:: det(vecteur v1){ return v1.x * y - x *v1.y;}
    int main(){
        vecteur v1(1,2),v2(0.5,4);
        cout<<"Le vecteur 1 : "<<endl;
        v1.affiche();
        cout<<"Le vecteur 2 : "<<endl;
        v2.affiche();
        cout<<"Les valeurs nouvelles de la vecteur 1 : "<<endl;
        v1=*v1.homotethie(2);
        v1.affiche();
        cout<< "Le determinant des deux vecteurs v1 et v2 : \n";
        cout <<"det(v1,v2) = "<<v1.det(v2);
    }

```

✓ L'exécution de programme donne :



```

C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP3_CPP_ZRAID...
Le vecteur 1 :
x = 1 y= 2
Le vecteur 2 :
x = 0.5 y= 4
Les valeurs nouvelles de la vecteur 1 :
x = 2 y= 4
Le determinant des deux vecteurs v1 et v2 :
det(v1,v2) = -6
-----
Process exited after 0.1479 seconds with return value 0
Appuyez sur une touche pour continuer...

```

Exercice c:

```

#include<iostream>
#include<conio.h>
using namespace std;

class vecteur
{
    float x,y;

```

```

    public :
        vecteur(float,float);
        vecteur &homotethie(float);
        float det(vecteur);
        void affiche();
};

vecteur :: vecteur(float abs=0.,float ord =0.)
{
    x=abs;
    y=ord;
}

vecteur &vecteur :: homotethie(float val)
{
    static vecteur res;
    res.x=x*val;
    res.y=y*val;
    return res;
}

void vecteur :: affiche()
{
    cout<<"x = "<<x<<" y= "<<y<<endl;
}

float vecteur:: det(vecteur v1)
{
    return v1.x * y - x *v1.y;
}

int main()
{
    vecteur v1(1,2),v2(0.5,4);
    cout<<"Le vecteur 1 : "<<endl;
    v1.affiche();
    cout<<"Le vecteur 2 : "<<endl;
    v2.affiche();
    cout<<"Les valeurs nouvelles de la vecteur 1 : "<<endl;
    v1=v1.homotethie(2);
    v1.affiche();
    cout<< "Le determinant des deux vecteurs v1 et v2 : \n";
    cout <<"det(v1,v2) = "<<v1.det(v2);
}

```

✓ L'exécution de programme donne :

```
C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP3_CPP_ZRAID...
Le vecteur 1 :
x = 1 y= 2
Le vecteur 2 :
x = 0.5 y= 4
Les valeurs nouvelles de la vecteur 1 :
x = 2 y= 4
Le determinant des deux vecteurs v1 et v2 :
det(v1,v2) = -6
-----
Process exited after 0.1479 seconds with return value 0
Appuyez sur une touche pour continuer...
```

Fonction membres statiques :

Exemple :

```
#include<iostream>
#include<conio.h>
using namespace std;

class compte_objet
{
    static int ctr;
    public :
        compte_objet();
        ~compte_objet();
        static void compte();
};
int compte_objet ::ctr=0;
compte_objet :: compte_objet(){
    cout<< "++construction : il y a maintenant "<<++ctr<<" objets\n";
}
compte_objet :: ~compte_objet(){
    cout<< "--desruccion : il y a maintenant "<<--ctr<<" objets\n";
}
void compte_objet :: compte(){
    cout <<"Appel compte : il y a "<< ctr <<" objets\n";
}
void fonction()
{
    compte_objet u,v;
}
int main(){
    void fonction();
    compte_objet :: compte();
```

```

compte_objet a;
compte_objet :: compte();
fonction();
compte_objet :: compte();
compte_objet b;
compte_objet :: compte();
}

```

✓ L'exécution de programme donne :

```

C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP3_C...
Appel compte : il y a 0 objets
++construction : il y a maintenant 1 objets
Appel compte : il y a 1 objets
++construction : il y a maintenant 2 objets
++construction : il y a maintenant 3 objets
--desruction : il y a maintenant 2 objets
--desruction : il y a maintenant 1 objets
Appel compte : il y a 1 objets
++construction : il y a maintenant 2 objets
Appel compte : il y a 2 objets
--desruction : il y a maintenant 1 objets
--desruction : il y a maintenant 0 objets

-----
Process exited after 0.1594 seconds with return value 0
Appuyez sur une touche pour continuer...

```

Mot clé « This » :

Exercice 1:

```

#include<iostream>
#include<conio.h>
using namespace std;
class point
{
    int x,y;
    public :
    point(int abs=0,int ord=0) //constructeur
    {
        x=abs;
        y=ord;
    }
    void affiche();
};
void point :: affiche(){

```

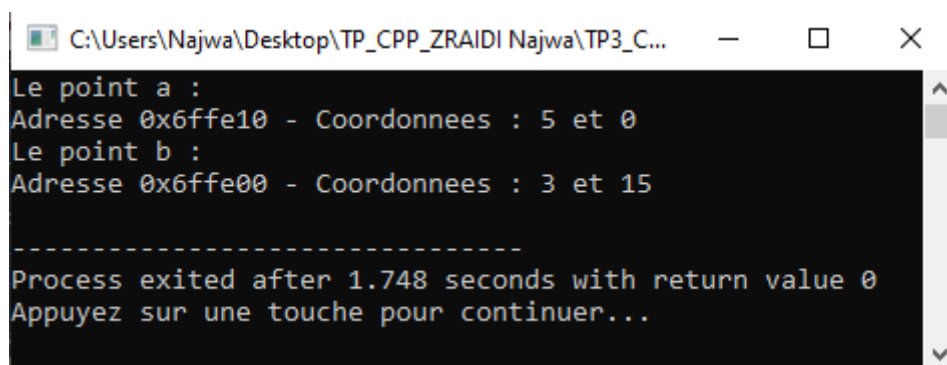
```

        cout<<"Adresse "<<this<<" - Coordonnees : "<<x<<" et "<<y<<"\n";
    }

int main(){
    point a(5),b(3,15);
    cout<<"Le point a : \n";
    a.affiche();
    cout<<"Le point b :\n";
    b.affiche();
    getch();
}

```

✓ L'exécution de programme donne :



```

C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP3_C...
Le point a :
Adresse 0x6ffe10 - Coordonnees : 5 et 0
Le point b :
Adresse 0x6ffe00 - Coordonnees : 3 et 15

-----
Process exited after 1.748 seconds with return value 0
Appuyez sur une touche pour continuer...

```

Exercise 2:

```

#include<iostream>
#include<conio.h>
using namespace std;
class point
{
    int x,y;
    public :
    point(int abs=0,int ord=0) //constructeur
    {
        x=abs;
        y=ord;
    }
    void affiche();
    int coincide(point *);
};

void point :: affiche(){
    cout<<"Adresse "<<this<<" - Coordonnees : "<<x<<" et "<<y<<"\n";
}

int point :: coincide(point *pt){

    if((pt->x==this->x)&&(pt->y==this->y))

```

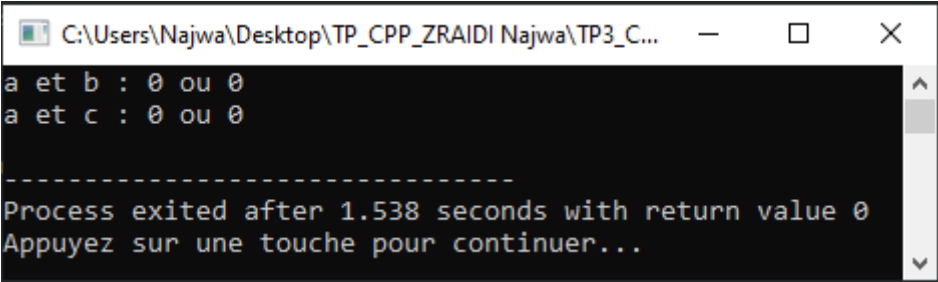
```

    /*pt.x acces d'absce de point
    pt et pt.x acces d'ordonnee de point pt*/
    return 1;
    else
    return 0;
}

int main(){
    int test1,test2;
    point a,b(1),c(0,2);
    test1=a.coincide(&b);
    test2=b.coincide(&a);
    cout<<"a et b : "<<test1 <<" ou "<<test2<<endl;
    test1=a.coincide(&c);
    test2=c.coincide(&a);
    cout<<"a et c : "<<test1 <<" ou "<<test2<<endl;
    getch();
}

```

✓ L'exécution de programme donne :



```

C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP3_C...
a et b : 0 ou 0
a et c : 0 ou 0

-----
Process exited after 1.538 seconds with return value 0
Appuyez sur une touche pour continuer...

```

Exercice 3:

```

#include<iostream>
#include<conio.h>
using namespace std;

class vecteur
{
    float x,y;
    public :
        vecteur(float,float);
        float prod_scal(vecteur);
        vecteur somme(vecteur);
        void affiche();
};

vecteur :: vecteur(float abs=0.,float ord =0.){
    x=abs;
    y=ord;
}

```

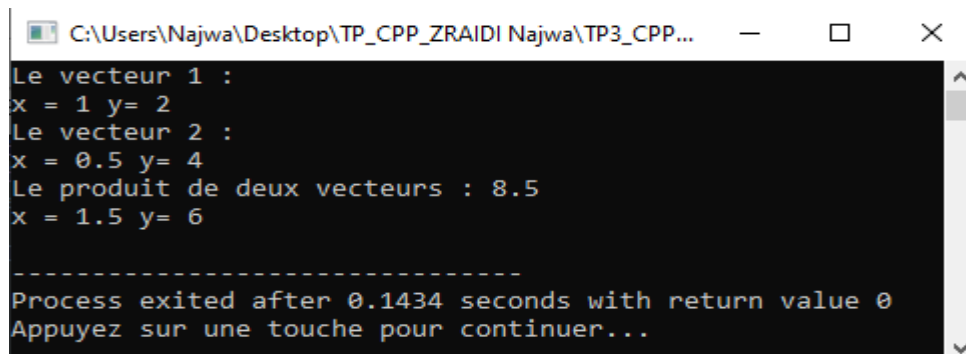


```

}
float vecteur :: prod_scal(vecteur v1){
    return this->x*v1.x + this->y * v1.y;
}
vecteur vecteur :: somme(vecteur v1){
    vecteur res;
    res.x = this->x + v1.x;
    res.y = this->y + v1.y;
    return res;
}
void vecteur :: affiche(){
    cout<<"x = "<<x<<" y= "<<y<<endl;
}
int main(){
    vecteur v1(1,2),v2(0.5,4),res;
    cout<<"Le vecteur 1 : "<<endl;
    v1.affiche();
    cout<<"Le vecteur 2 : "<<endl;
    v2.affiche();
    cout<< "Le produit de deux vecteurs : "<< v1.prod_scal(v2)<<endl;
    res=v1.somme(v2);
    res.affiche(); }

```

✓ L'exécution de programme donne :



```

C:\Users\Najwa\Desktop\TP_CPP_ZRA\DI Najwa\TP3_CPP...
Le vecteur 1 :
x = 1 y= 2
Le vecteur 2 :
x = 0.5 y= 4
Le produit de deux vecteurs : 8.5
x = 1.5 y= 6

-----
Process exited after 0.1434 seconds with return value 0
Appuyez sur une touche pour continuer...

```