

Travaux Pratiques

COURS



🚦 Ce TP réalisé par :

ZRAIDI NAJWA
(GLSID1)

Construction et destruction des objets automatiques :

Exemple :

- ✓ Ce programme permet d'étudier soigneusement à quel moment sont créés puis détruits les objets déclarés .

```
#include<iostream>
using namespace std;

class point {
    int x,y;
    public:
        point(int,int);
        ~point();
};

point :: point(int abs,int ord)
{
    x=abs;y=ord;
    cout<<"Construction du point "<<x<<" . "<<y<<endl;
}

point :: ~point()
{
    cout<<"Destruction du point "<<x<<" "<<y<<endl;
}

void test()
{
    cout<< "Debut de test ()"<<endl;
    point u(3,7);
    cout<<"Fin de test () \n ";
}

int main(){
    cout<<"Debut de main()\n";
    point a(1,4);
    test();
    point b(5,10);
    for(int i=0;i<3;i++)
    {
        cout<<" Boucle tour numero "<<i<<"\n";
        point(7+i,12+i);} cout<<"Fin de main() \n ";
}
```

- ✓ L'exécution de programme donne :

```
C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP4_CPP...
Debut de main()
Construction du point 1 . 4
Debut de test ()
Construction du point 3 . 7
Fin de test ()
  Destruction du point 3 7
Construction du point 5 . 10
  Boucle tour numero 0
Construction du point 7 . 12
Destruction du point 7 12
  Boucle tour numero 1
Construction du point 8 . 13
Destruction du point 8 13
  Boucle tour numero 2
Construction du point 9 . 14
Destruction du point 9 14
Fin de main()
  Destruction du point 5 10
Destruction du point 1 4

-----
Process exited after 0.07157 seconds with return value 0
Appuyez sur une touche pour continuer...
```

Construction et destruction des objets statiques :

Exemple :

- ✓ L'objectif de cet exemple est d'étudier soigneusement à quel moment sont créés puis détruits les objets déclarés .

```
#include<iostream>
using namespace std;

class point {
    int x,y;
    public:
        point(int,int);
        ~point();
};

point :: point(int abs,int ord){
    x=abs;y=ord;
    cout<<"Construction du point "<<x<<" "<<y<<endl;
}

point :: ~point(){
    cout<<"Destruction du point "<<x<<" "<<y<<endl;
}

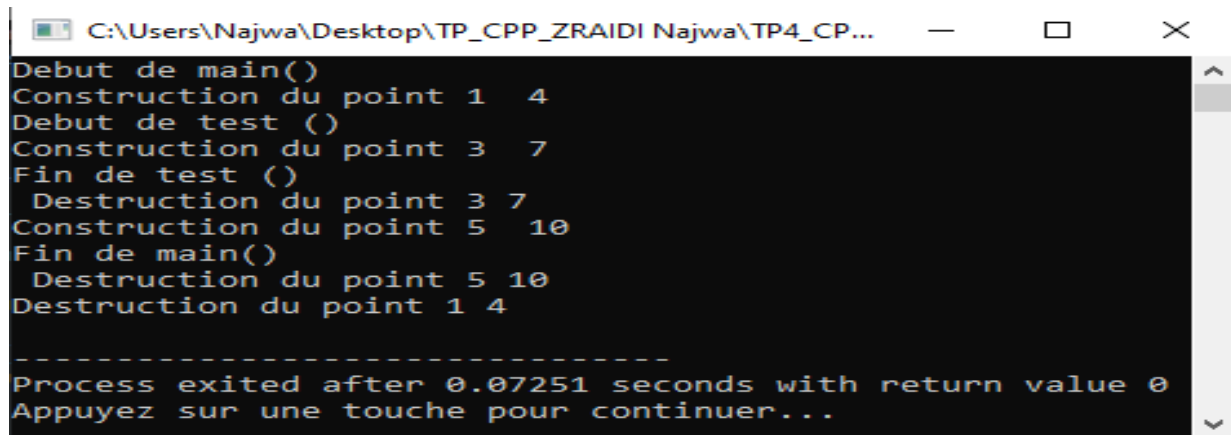
void test(){
    cout<< "Debut de test ()"<<endl;
    point u(3,7);
    cout<<"Fin de test () \n ";
}
```

```

}
int main(){
    cout<<"Debut de main()\n";
    point a(1,4);
    test();
    point b(5,10);
    cout<<"Fin de main() \n ";
}

```

- ✓ L'exécution de programme donne :



```

C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP4_CP...
Debut de main()
Construction du point 1 4
Debut de test ()
Construction du point 3 7
Fin de test ()
Destruction du point 3 7
Construction du point 5 10
Fin de main()
Destruction du point 5 10
Destruction du point 1 4

-----
Process exited after 0.07251 seconds with return value 0
Appuyez sur une touche pour continuer...

```

Construction et destruction des objets globaux :

Exemple :

- ✓ L'objectif de cet exemple est d'étudier soigneusement à quel moment sont créés puis détruits les objets déclarés .

```

#include<iostream>
using namespace std;

class point {
    int x,y;
    public:
        point(int,int);
        ~point();
};

point :: point(int abs,int ord){
    x=abs;y=ord;
    cout<<"Construction du point "<<x<<" "<<y<<endl;
}

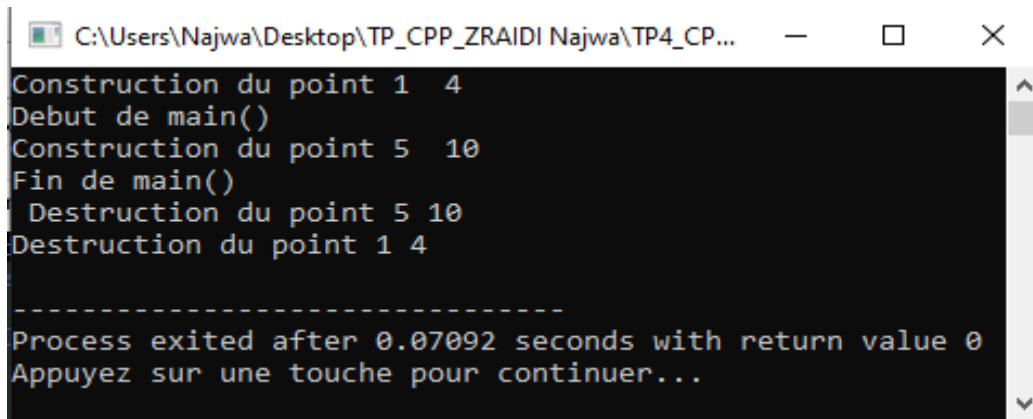
point :: ~point(){
    cout<<"Destruction du point "<<x<<" "<<y<<endl;
}

point a(1,4); // variable globale

```

```
int main(){
    cout<<"Debut de main()\n";
    point b(5,10);
    cout<<"Fin de main() \n ";
}
```

✓ L'exécution de programme donne :



```
C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP4_CP...
Construction du point 1 4
Debut de main()
Construction du point 5 10
Fin de main()
Destruction du point 5 10
Destruction du point 1 4

-----
Process exited after 0.07092 seconds with return value 0
Appuyez sur une touche pour continuer...
```

Construction et destruction des objets temporaires :

Exemple :

```
#include<iostream>
using namespace std;

class point {
    int x,y;
public:
    point(int,int);
    ~point();
};

point :: point(int abs,int ord){
    x=abs;y=ord;
    cout<<"Construction du point "<<x<<" "<<y<<" a l'adresse : "<<this<<endl;
}

point :: ~point(){
    cout<<"Destruction du point "<<x<<" "<<y<<" a l'adresse : "<<this<<endl;
}

void test(){
    cout<<"Debut de test ()"<<endl;
    point u(3,7);
    cout<<"Fin de test () \n ";
}

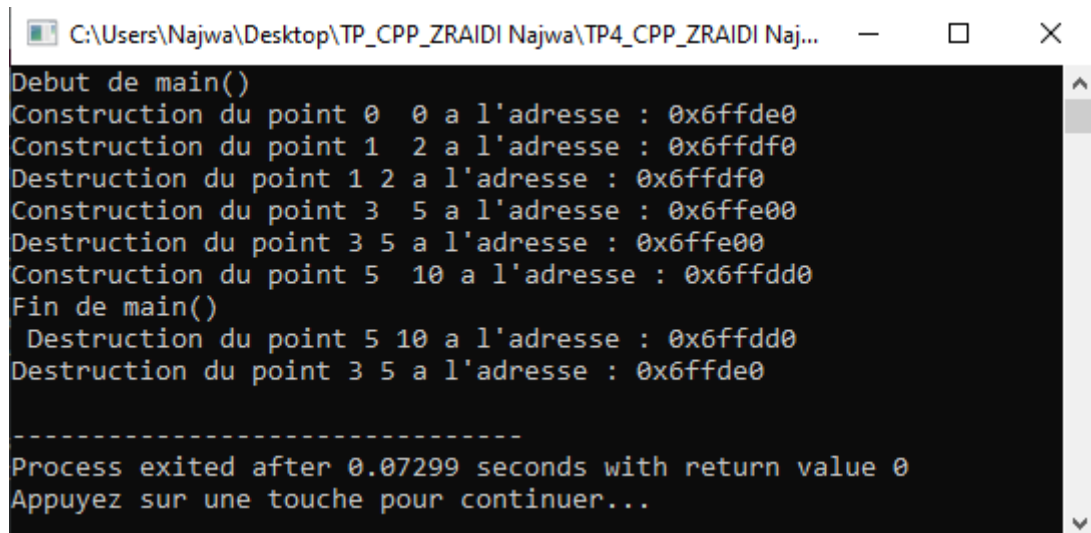
int main(){
    cout<<"Debut de main()\n";
```

```

point a(0,0);
a=point(1,2);
a=point(3,5);
point b(5,10);
cout<<"Fin de main() \n ";
}

```

✓ L'exécution de programme donne :



```

C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP4_CPP_ZRAIDI Naj...
Debut de main()
Construction du point 0 0 a l'adresse : 0x6ffde0
Construction du point 1 2 a l'adresse : 0x6ffdf0
Destruction du point 1 2 a l'adresse : 0x6ffdf0
Construction du point 3 5 a l'adresse : 0x6ffe00
Destruction du point 3 5 a l'adresse : 0x6ffe00
Construction du point 5 10 a l'adresse : 0x6ffdd0
Fin de main()
Destruction du point 5 10 a l'adresse : 0x6ffdd0
Destruction du point 3 5 a l'adresse : 0x6ffde0

-----
Process exited after 0.07299 seconds with return value 0
Appuyez sur une touche pour continuer...

```

Construction et destruction des objets dynamiques :

Exemple :

```

#include<iostream>
using namespace std;

class point {
    int x,y;
public:
    point(int,int);
    ~point();
};

point :: point(int abs,int ord){
    x=abs;y=ord;
    cout<<"Construction du point "<<x<<" "<<y<<endl;
}

point :: ~point(){
    cout<<"Destruction du point "<<x<<" "<<y<<endl;
}

int main(){
    void fct(point *);
    point *adr;

```

```

    cout<<"Debut de main()\n";
    adr = new point(3,7);//reservation de place en memoire
    fct(adr);
    delete adr; //liberation de la place
    cout<<"Fin de main() \n ";
}
void fct(point *adp){
    cout <<"Debut de la fonction \n";
    delete adp; //liberation de la place
    cout<<"Fin de la fonction \n ";
}

```

✓ L'exécution de programme donne :

```

C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP4_CPP_ZRAIDI Naj...
Debut de main()
Construction du point 3 7
Debut de la fonction
Destruction du point 3 7
Fin de la fonction
Destruction du point 858832 0

-----
Process exited after 0.5636 seconds with return value 3221226356
Appuyez sur une touche pour continuer...

```

Initialisation des objets :

Exemple :

```

#include<iostream>
using namespace std;

class point {
    int x,y;
public:
    point(int,int);
    ~point();
};

point :: point(int abs,int ord){
    x=abs;y=ord;
    cout<<"Construction du point "<<x<<" "<<y<<" a l'adresse : "<<this<<endl;
}

point :: ~point(){
    cout<<"Destruction du point "<<x<<" "<<y<<" a l'adresse : "<<this<<endl;
}

```

```

}

int main(){
    cout<<"Debut de main()\n";
    point a(3,7);
    point b=a;
    cout<<"Fin de main() \n ";
}

```

✓ L'exécution de programme donne :

```

C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP4_CPP_ZRAI...
Debut de main()
Construction du point 3 7 a l'adresse : 0x6ffe00
Fin de main()
Destruction du point 3 7 a l'adresse : 0x6ffdf0
Destruction du point 3 7 a l'adresse : 0x6ffe00

-----
Process exited after 0.06654 seconds with return value 0
Appuyez sur une touche pour continuer...

```

Exemple 1:

```

#include<iostream>
using namespace std;

class liste
{
    int taille;
    float *adr;
public :
    liste(int);
    ~liste();
};

liste :: liste(int t){
    taille =t;
    adr= new float[taille];
    cout<<"Construction : \n";
    cout<<" - Adresse de l'objet : "<<this<<endl;
    cout<<" - Adresse de liste : "<<adr<<endl;
}

liste :: ~liste(){
    cout<<"Destruction : \n - Adresse de l'objet : "<<this<<endl;
    cout<<" - Adresse de liste : "<<adr<<endl;
    delete adr;
}

int main(){
    cout<<"Debut de main()"<<endl;

```



```

liste a(3);
liste b=a;
cout<<"Fin de main\n";
}

```

✓ L'exécution de programme donne :

```

C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP4_CPP_ZRAIDI Najwa\C...
Debut de main()
Construction :
- Adresse de l'objet : 0x6ffe00
- Adresse de liste : 0x8115b0
Fin de main
Destruction :
- Adresse de l'objet : 0x6ffdf0
- Adresse de liste : 0x8115b0
Destruction :
- Adresse de l'objet : 0x6ffe00
- Adresse de liste : 0x8115b0

-----
Process exited after 0.4884 seconds with return value 3221226356
Appuyez sur une touche pour continuer...

```

Exemple 2:

```

#include<iostream>
using namespace std;

class liste
{
    int taille;
    float *adr;
public :
    liste(int);
    liste(liste &);
    ~liste();
};

liste :: liste(int t){
    taille =t;
    adr= new float[taille];
    cout<<"Construction : \n";
    cout<<" - Adresse de l'objet : "<<this<<endl;
    cout<<" - Adresse de liste : "<<adr<<endl;
}

liste :: liste(liste &v){ //passage par reference obligatoire
    taille =v.taille;
    adr= new float[taille];
    for(int i=0;i<taille;i++)
        adr[i]=v.adr[i];
    cout<<"Construction par recopie : \n";
    cout<<" - Adresse de l'objet : "<<this<<endl;
}

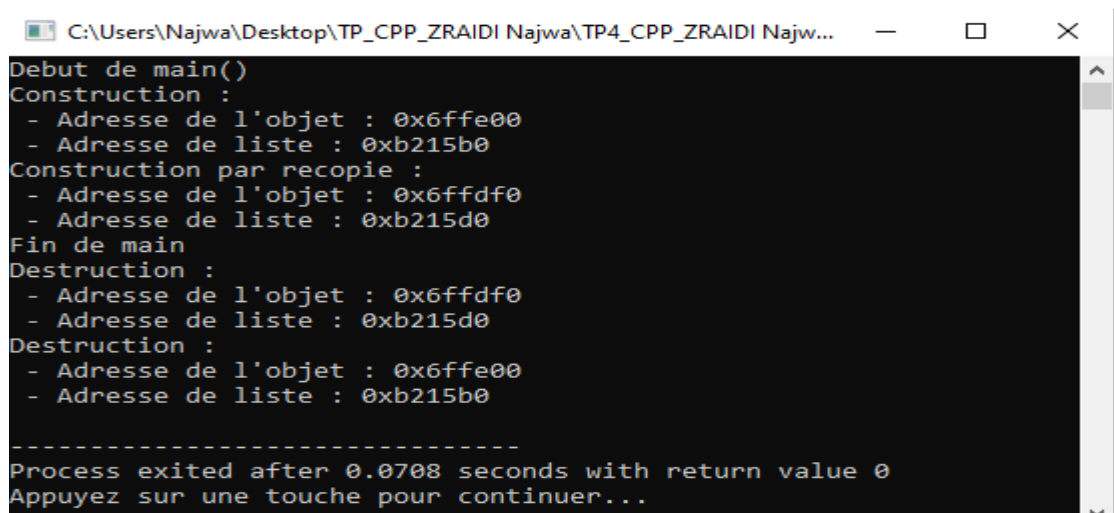
```

```

        cout<<" - Adresse de liste : "<<adr<<endl;
    }
    liste::~liste(){
        cout<<"Destruction : \n - Adresse de l'objet : "<<this<<endl;
        cout<<" - Adresse de liste : "<<adr<<endl;
        delete adr;
    }
    int main(){
        cout<<"Debut de main()"<<endl;
        liste a(3);
        liste b=a;
        cout<<"Fin de main\n";
    }

```

✓ L'exécution de programme donne :



```

C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP4_CPP_ZRAIDI Najw...
Debut de main()
Construction :
- Adresse de l'objet : 0x6ffe00
- Adresse de liste : 0xb215b0
Construction par recopie :
- Adresse de l'objet : 0x6ffdf0
- Adresse de liste : 0xb215d0
Fin de main
Destruction :
- Adresse de l'objet : 0x6ffdf0
- Adresse de liste : 0xb215d0
Destruction :
- Adresse de l'objet : 0x6ffe00
- Adresse de liste : 0xb215b0
-----
Process exited after 0.0708 seconds with return value 0
Appuyez sur une touche pour continuer...

```

Rôle du constructeur lorsqu'une fonction retourne un objet :

Exemple 1 :

```

#include<iostream>
using namespace std;

class point {
    int x,y;
    public:
        point(int,int);
        point(point &);//constructeur par recopie
        point symetrique();

```

```

        void affiche(){ cout<<"x= "<<x<<" y= "<<y<<"\n";    }
        ~point();
};
point :: point(int abs=0,int ord=0){
    x=abs;y=ord;
    cout<<"Construction du point "<<x<<" "<<y<<" a l'adresse : "<<this<<endl;
}
point :: point(point &pt){
    x =pt.x;y=pt.y;
    cout<<"Construction par recopie du point "<<x<<" "<<y<<" a l'adresse
: "<<this<<endl;
}
point point :: symetrique(){
    point res;
    cout<<"*****\n";
    res.x=-x;
    res.y=-y;
    cout<<"#####\n";
    return res;
}
point :: ~point(){
    cout<<"Destruction du point "<<x<<" "<<y<<" a l'adresse : "<<this<<endl;
}
int main(){
    cout<<"Debut de main()\n";
    point a(1,4),b;
    cout<<" Avant appel a symetrique  : \n";
    b=a.symetrique();
    b.affiche();
    cout<<" Apres appel a symetrique  : \n";
    cout<<"Fin de main() \n ";}

```

✓ L'exécution de programme donne :

```

C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP4_CPP_ZRAI...
Debut de main()
Construction du point 1 4 a l'adresse : 0x6ffdf0
Construction du point 0 0 a l'adresse : 0x6ffde0
Avant appel a symetrique :
Construction du point 0 0 a l'adresse : 0x6ffe00
*****
#####
Destruction du point -1 -4 a l'adresse : 0x6ffe00
x= -1 y= -4
Apres appel a symetrique :
Fin de main()
Destruction du point -1 -4 a l'adresse : 0x6ffde0
Destruction du point 1 4 a l'adresse : 0x6ffdf0

-----
Process exited after 0.07741 seconds with return value 0
Appuyez sur une touche pour continuer...

```

Exemple 2 :

```
#include<iostream>
using namespace std;

class liste
{
    int taille;
    float *adr;
public :
    liste(int);
    liste(liste &);
    void saisie();
    void affiche();
    liste oppose();
    ~liste();
};

liste :: liste(int t){
    taille =t;
    adr= new float[taille];
    cout<<"Construction : \n";
    cout<<" - Adresse de l'objet : "<<this<<endl;
    cout<<" - Adresse de liste : "<<adr<<endl;
}

liste :: liste(liste &v){ //passage par r f rence obligatoire
    taille =v.taille;
    adr= new float[taille];
    for(int i=0;i<taille;i++)
        adr[i]=v.adr[i];
    cout<<"Construction par recopie : \n";
    cout<<" - Adresse de l'objet : "<<this<<endl;
    cout<<" - Adresse de liste : "<<adr<<endl;
}

liste :: ~liste(){
    cout<<"Destruction : \n - Adresse de l'objet : "<<this<<endl;
    cout<<" - Adresse de liste : "<<adr<<endl;
    delete adr;
}

void liste :: saisie(){
    int i;
    for(int i=0;i<taille;i++)
    {
        cout<<"Entrer un nombre : \n";
        cin>>*(adr+i);
    }
}

void liste :: affiche(){
```

```

int i;
for(int i=0;i<taille;i++)
{
    cout<<*(adr+i)<<" adresse de l'objet : "<<this<<" Adresse de liste : 
"<<adr<<endl;  }}
liste liste :: oppose(){
    liste res(taille);
    cout<<"liste oppose : \n";
    for(int i=0;i<taille;i++)
        res.adr[i]= -adr[i];
    for(int i=0;i<taille;i++)
        cout<<res.adr[i]<<" \n";
    return res;}
int main(){
    cout<<"Debut de main()"<<endl;
    liste a(3),b(3);
    a.saisie();
    a.affiche();
    b=a.oppose();
    b.affiche();cout<<"Fin de main\n";}

```

✓ L'exécution de programme donne :

```

C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP4_CPP_ZRAIDI Najwa\Cours\Exemple 2(Role du constru
Debut de main()
Construction :
- Adresse de l'objet : 0x6ffdf0
- Adresse de liste : 0xa215f0
Construction :
- Adresse de l'objet : 0x6ffde0
- Adresse de liste : 0xa21610
Entrer un nombre :
4
Entrer un nombre :
2
Entrer un nombre :
1
4 adresse de l'objet : 0x6ffdf0 Adresse de liste : 0xa215f0
2 adresse de l'objet : 0x6ffdf0 Adresse de liste : 0xa215f0
1 adresse de l'objet : 0x6ffdf0 Adresse de liste : 0xa215f0
Construction :
- Adresse de l'objet : 0x6ffe00
- Adresse de liste : 0xa21630
liste oppose :
-4
-2
-1
Destruction :
- Adresse de l'objet : 0x6ffe00
- Adresse de liste : 0xa21630
1.48875e-038 adresse de l'objet : 0x6ffde0 Adresse de liste : 0xa21630
0 adresse de l'objet : 0x6ffde0 Adresse de liste : 0xa21630
1.48778e-038 adresse de l'objet : 0x6ffde0 Adresse de liste : 0xa21630
Fin de main
Destruction :
- Adresse de l'objet : 0x6ffde0
- Adresse de liste : 0xa21630
-----
Process exited after 5.287 seconds with return value 3221226356
Appuyez sur une touche pour continuer...

```

Les tableaux d'objets :

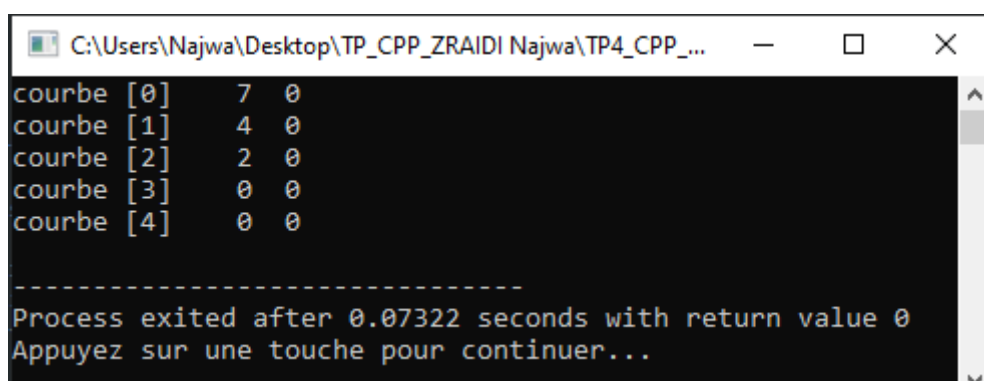
Exemple :

```
#include<iostream>
using namespace std;

class point {
    int x,y;
    public:
        point(int abs=0,int ord=0)
        {
            x=abs;y=ord;
        }
        void affiche(){
            static int i=0;
            cout<<"courbe ["<<i<<"    "<<x<<"    "<<y<<endl;
            i++;
        }
};

int main(){
    point  courbe[5]={7,4,2};
    for (int i=0;i<5;i++){
        courbe[i].affiche();
    }
}
```

✓ L'exécution de programme donne :



```
C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP4_CPP_...
courbe [0]    7    0
courbe [1]    4    0
courbe [2]    2    0
courbe [3]    0    0
courbe [4]    0    0

-----
Process exited after 0.07322 seconds with return value 0
Appuyez sur une touche pour continuer...
```

Objets membres ou objets d'objets :

Exemple :

```
#include<iostream>
```

```

using namespace std;

class point {
    int x,y;
    public:
        point(int abs=0,int ord=0)
        {
            x=abs;y=ord;
            cout<<"Construction du point "<<x<<" "<<y<<endl;
        }
};

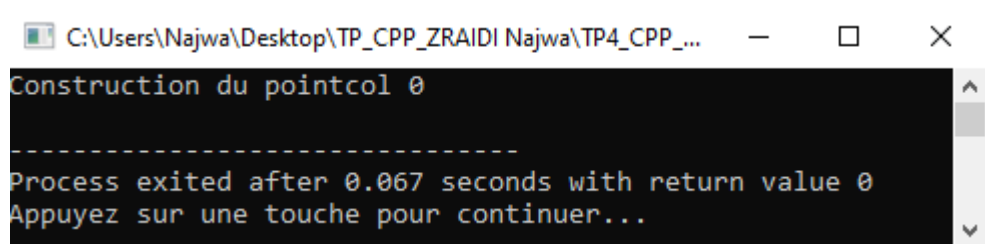
class pointcol{
    point p;
    int couleur;
    public :
        pointcol(int,int,int);
};

pointcol ::pointcol(int abs,int ord ,int coul) :p(abs,ord)
{
    couleur = coul ;
    cout<<"Construction du pointcol "<<coul<<endl;
}

int main()
{
    pointcol a(1,3,0);
}

```

✓ L'exécution de programme donne :



```

C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP4_CPP_...
Construction du pointcol 0
-----
Process exited after 0.067 seconds with return value 0
Appuyez sur une touche pour continuer...

```