



L'Ecole Normale Supérieure de l'Enseignement  
Technique de Mohammedia

# COMPTE RENDU DES TRAVAUX PRATIQUES DE LA CHAPITRE 3

Propriétés des fonctions membres

09/04/2022

Préparé par  
ACHRAF TAFFAH

Encadré par  
PR. KHALIFA MANSOURI

# TABLE DES MATIERES

<b>introduction</b> .....	3
<b>Partie pratique</b> .....	4
<b>1. Exemples de cour :</b> .....	4
EXEMPLE 1 : .....	4
Surdéfinition des fonctions membres .....	4
EXEMPLE 2 : .....	5
Fonctions membres en ligne .....	5
EXEMPLE 3 : .....	6
EXEMPLE 4 : .....	7
Initialisation des paramètres par défaut .....	7
EXEMPLE 5 : .....	8
Exercice 3 : Classe "Vector" .....	8
EXEMPLE 6 : .....	9
Objets retourné par une fonction membre .....	9
EXEMPLE 8 : .....	11
EXEMPLE 9 : .....	12
Fonctions membres statiques .....	12
EXEMPLE 10 : .....	13
Le mot clé "this" .....	13
<b>2. Exercices :</b> .....	13
Exercice 1 & 2 : .....	13
<b>conclusion</b> .....	14

# INTRODUCTION

UNE FONCTION EST UN ENSEMBLE D'INSTRUCTIONS QUI PRENNENT DES ENTREES, EFFECTUENT DES CALCULS SPECIFIQUES ET PRODUISENT UNE SORTIE.

L'IDEE EST DE REGROUPER CERTAINES TACHES EFFECTUEES DE MANIERE COURANTE OU REPETEE ET DE CREER UNE FONCTION PERMETTANT D'APPELER LA FONCTION AU LIEU D'ECRIRE LE MEME CODE ENCORE ET ENCORE POUR DIFFERENTES ENTREES.

UNE DECLARATION DE FONCTION INDIQUE AU COMPILATEUR LE NOM, LE TYPE DE RETOUR ET LES PARAMETRES D'UNE FONCTION.

UNE DEFINITION DE FONCTION FOURNIT LE CORPS REEL DE LA FONCTION.

# PARTIE PRATIQUE

## 1. Exemples de cour :

### EXEMPLE 1 :

#### SURDEFINITION DES FONCTIONS MEMBRES

```
1 #include <iostream>
2
3 class Point
4 {
5 private:
6     int x;
7     int y;
8
9 public:
10    Point();
11    Point(int);
12    Point(int, int);
13    void afficher() const;
14    void afficher(std::string) const;
15
16    Point::Point(){
17        x = 0; y = 0;
18    }
19    Point::Point(int abs){
20        x = abs; y = abs;
21    }
22    Point::Point(int abs, int ord){
23        x = abs; y = ord;
24    }
25    void Point::afficher() const{
26        std::cout << "(" << x << ", " << y << ")" << std::endl;
27    }
28    void Point::afficher(std::string message) const{
29        std::cout << message; this->afficher();
30    }
31
32 int main(int argc, char const *argv[]){
33     Point p1; p1.afficher();
34     Point p2(5); p2.afficher("p2 : ");
35     Point p3(5, 6); p3.afficher("p3 : "); std::cin.get(); return 0;
36 }
```

Voilà l'affichage après l'exécution de ce programme.

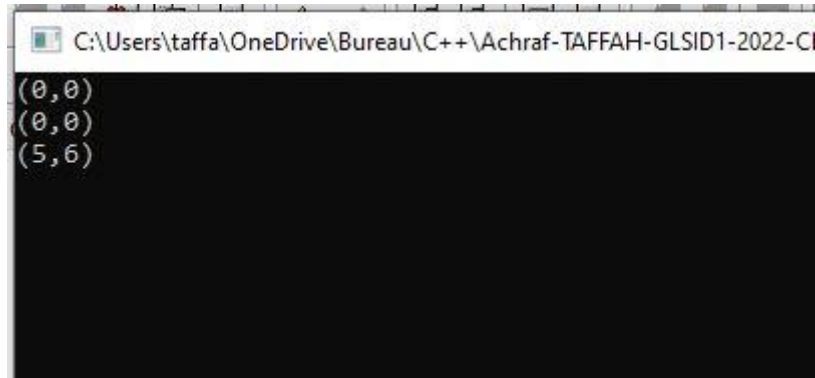
```
C:\Users\taffa\OneDrive\Bureau\C++\Achraf-TAFFAH-GLSID1-2022-CHAPITRE3
(0,0)
p2 : (5,5)
p3 : (5,6)
```

## EXEMPLE 2 :

### FONCTIONS MEMBRES EN LIGNE

```
exemple5.cpp  exemple6.cpp  exemple7.cpp  exemple8.cpp  exemple9.cpp  exemple10.cpp  [*] exemple2.cpp  exemple
1  #include <iostream>
2
3  class Point
4  {
5  private:
6      int x;
7      int y;
8
9  public:
10     Point() { x = 0; y = 0; }
11     Point(int abs) { x = abs; y = abs; }
12     Point(int abs, int ord) { x = abs; y = ord; }
13     void afficher() const;
14 };
15
16 void Point::afficher() const
17 {
18     std::cout << "(" << x << "," << y << ")" << std::endl;
19 }
20
21 int main(int argc, char const *argv[])
22 {
23     Point p1, p2;
24     p1.afficher();
25     p2.afficher();
26     Point p3(5, 6);
27     p3.afficher();
28     std::cin.get();
29     return 0;
30 }
```

Voilà l’affichage après l’exécution de ce programme.5

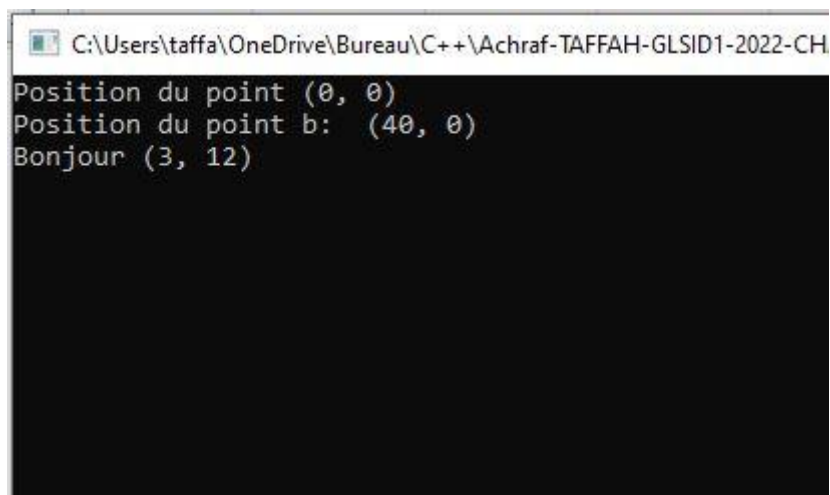


```
C:\Users\taffa\OneDrive\Bureau\C++\Achraf-TAFFAH-GLSID1-2022-CI
(0,0)
(0,0)
(5,6)
```

### EXEMPLE 3 :

```
exemple5.cpp  exemple6.cpp  exemple7.cpp  exemple8.cpp  exemple9.cpp  exemple10.cpp  exemple3.cpp  exemple4
1  #include <iostream>
2
3  class Point
4  {
5  private:
6      int x, y;
7
8  public:
9      Point(int abs = 0, int ord = 0) : x(abs), y(ord) {}
10     void affiche(std::string = "Position du point") const;
11 };
12
13 void Point::affiche(std::string message) const
14 {
15     std::cout << message << " (" << x << ", " << y << ")" << std::endl;
16 }
17
18 int main(int argc, char const *argv[])
19 {
20     Point a, b(40);
21     a.affiche();
22     b.affiche("Position du point b: ");
23     char text[10] = "Bonjour";
24     Point c(3, 12);
25     c.affiche(text);
26     std::cin.get();
27     return 0;
28 }
```

Voilà l’affichage après l’exécution de ce programme.6



```
C:\Users\taffa\OneDrive\Bureau\C++\Achraf-TAFFAH-GLSID1-2022-CH
Position du point (0, 0)
Position du point b: (40, 0)
Bonjour (3, 12)
```

### Comparaison

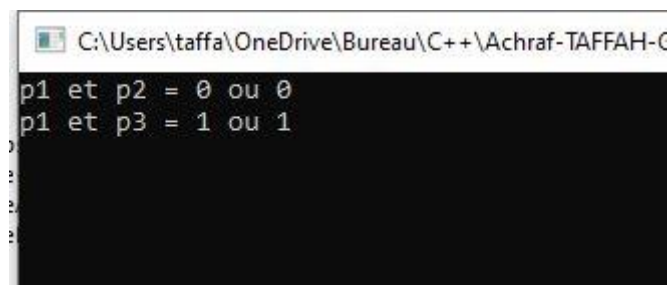
Le fichier objet généré du programme ayant la surdéfinition des fonctions membres prend plus de l’espace au mémoire que celle des fonctions membres en ligne.

## EXEMPLE 4 :

### INITIALISATION DES PARAMETRES PAR DEFAULT

```
exemple5.cpp exemple6.cpp exemple7.cpp exemple8.cpp exemple9.cpp exemple10.cpp [*] exemple4.cpp
1 #include <iostream>
2 class Point{
3 private:
4     int x;
5     int y;
6 public:
7     Point(int abs = 0, int ord = 0) { x = abs; y = ord; }
8     int coincide(Point) const;
9     int coincideAdresse(Point *) const;
10    int coincideRef(Point &) const;
11 };
12
13 int Point::coincide(Point p) const{
14     if (x == p.x && y == p.y)
15         return 1;
16     else
17         return 0;
18 }
19
20 int Point::coincideAdresse(Point *adpt) const {
21     if (x == adpt->x && y == adpt->y)
22         return 1;
23     else
24         return 0;
25 }
26
27 int Point::coincideRef(Point &ref) const{
28     if (x == ref.x && y == ref.y)
29         return 1;
30     else
31         return 0;
32 }
33
34 int main(int argc, char const *argv[]) {
35     int test1, test2; Point p1, p2(1), p3(0, 0);
36     test1 = p1.coincide(p2); test2 = p2.coincide(p1);
37     std::cout << "p1 et p2 = " << test1 << " ou " << test2 << std::endl; test1 = p1.coincide(p3);
38     test2 = p3.coincide(p1); std::cout << "p1 et p3 = " << test1 << " ou " << test2 << std::endl; std::cin.get();
39     return 0;
40 }
```

Voilà l'affichage après l'exécution de ce programme.7



```
C:\Users\taffa\OneDrive\Bureau\C++\Achraf-TAFFAH-C
p1 et p2 = 0 ou 0
p1 et p3 = 1 ou 1
```

## EXEMPLE 5 :

### EXERCICE 3 : CLASSE "VECTOR"

```
xemple5.cpp xemple6.cpp xemple7.cpp xemple8.cpp xemple9.cpp xemple10.cpp
#include <iostream>
class Vector{
private:
    float x, y;
public:
    Vector(float, float); void homothetie(float); Vector homothetie(float) const; void afficher() const; float determinant(Vector) const; float determinantAdresse(Vector *) const;
    float determinantRef(Vector &) const; float produit_scalaire(Vector) const; Vector somme(Vector) const;
};
Vector::Vector(float abs = 0, float ord = 0) : x(abs), y(ord) {}
void Vector::homothetie(float facteur){
    x *= facteur;
    y *= facteur;
}
// Par valeur
Vector Vector::homothetie(float facteur) const{
    Vector v(x * facteur, y * facteur);
    return v;
}
void Vector::afficher() const{
    std::cout << "(" << x << ", " << y << ")" << std::endl;
}
float Vector::determinant(Vector v) const{
    return x * v.y - y * v.x;
}
float Vector::determinantAdresse(Vector *v) const{
    return x * v->y - y * v->x;
}
float Vector::determinantRef(Vector &v) const{
    return x * v.y - y * v.x;
}
float Vector::produit_scalaire(Vector v) const{
    return x * v.x + y * v.y;
}
Vector Vector::somme(Vector v) const{
    return Vector(x + v.x, y + v.y);
}
```



## EXEMPLE 6 :

### OBJETS RETOURNE PAR UNE FONCTION MEMBRE

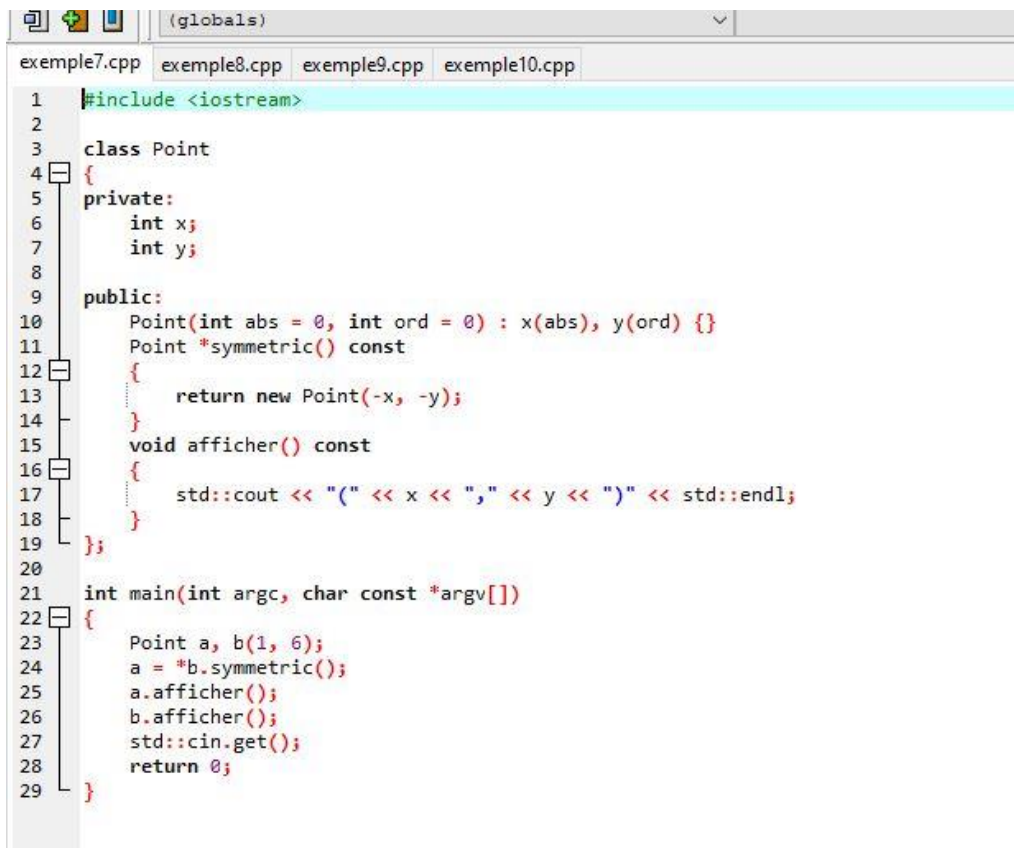
#### Retour par valeur

```
exemple6.cpp  exemple7.cpp  exemple8.cpp  exemple9.cpp  exemple10.cpp
1  #include <iostream>
2
3  class Point
4  {
5  private:
6      int x;
7      int y;
8
9  public:
10     Point(int abs = 0, int ord = 0) : x(abs), y(ord) {}
11     Point symmetric() const
12     {
13         return Point(-x, -y);
14     }
15     void afficher() const
16     {
17         std::cout << "(" << x << "," << y << ")" << std::endl;
18     }
19 };
20
21 int main(int argc, char const *argv[])
22 {
23     Point a, b(1, 6);
24     a = b.symmetric();
25     a.afficher();
26     b.afficher();
27     std::cin.get();
28     return 0;
29 }
```

Voilà l’affichage après l’exécution de ce programme.

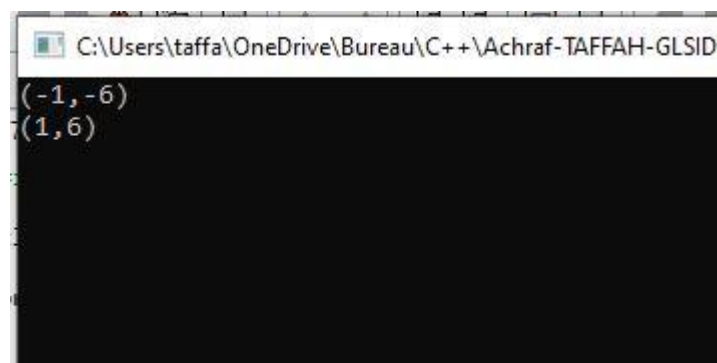
```
C:\Users\taffa\OneDrive\Bureau\C++\Achraf-TAFFAH-GLSID1-2022-CHAPITRE3\Cour\exemp
(-1,-6)
(1,6)
```

## Retour par adresse



```
1 #include <iostream>
2
3 class Point
4 {
5 private:
6     int x;
7     int y;
8
9 public:
10     Point(int abs = 0, int ord = 0) : x(abs), y(ord) {}
11     Point *symmetric() const
12     {
13         return new Point(-x, -y);
14     }
15     void afficher() const
16     {
17         std::cout << "(" << x << "," << y << ")" << std::endl;
18     }
19 };
20
21 int main(int argc, char const *argv[])
22 {
23     Point a, b(1, 6);
24     a = *b.symmetric();
25     a.afficher();
26     b.afficher();
27     std::cin.get();
28     return 0;
29 }
```

Voilà l’affichage après l’exécution de ce programme.10



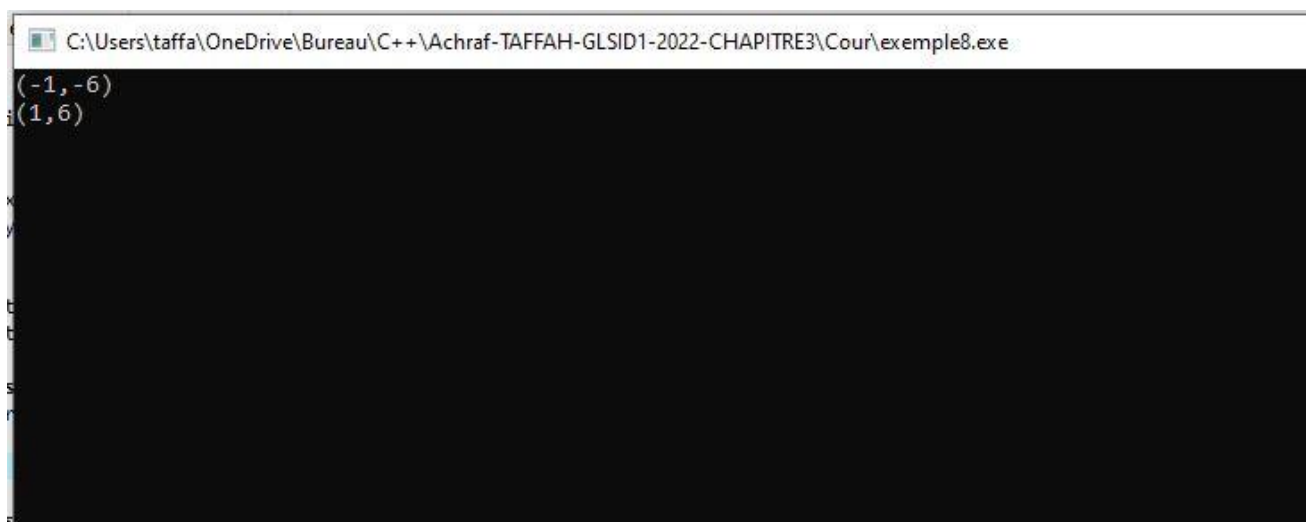
```
C:\Users\taffa\OneDrive\Bureau\C++\Achraf-TAFFAH-GLSID
(-1, -6)
(1, 6)
```

## EXEMPLE 8 :

### Retour par référence

```
exemple8.cpp  exemple9.cpp  exemple10.cpp
1  #include <iostream>
2
3  class Point
4  {
5  private:
6      int x;
7      int y;
8
9  public:
10     Point(int abs = 0, int ord = 0) : x(abs), y(ord) {}
11     Point &symmetric() const
12     {
13         static Point p(-x, -y);
14         return p;
15     }
16     void afficher() const
17     {
18         std::cout << "(" << x << "," << y << ")" << std::endl;
19     }
20 };
21
22 int main(int argc, char const *argv[])
23 {
24     Point a, b(1, 6);
25     a = b.symmetric();
26     a.afficher();
27     b.afficher();
28     std::cin.get();
29     return 0;
30 }
```

Voilà l’affichage après l’exécution de ce programme.11



```
C:\Users\taffa\OneDrive\Bureau\C++\Achraf-TAFFAH-GLSID1-2022-CHAPITRE3\Cour\exemple8.exe
(-1,-6)
(1,6)
```

## EXEMPLE 9 :

### FONCTIONS MEMBRES STATIQUES

```
1 #include <iostream>
2 class compte_objet
3 {
4     static int ctr;
5
6 public:
7     compte_objet();
8     ~compte_objet();
9     static void compte();
10 };
11 int compte_objet::ctr = 0;
12 compte_objet::compte_objet()
13 {
14     std::cout << "Construction: il y a maintenant " << ++ctr << " objets" << std::endl;
15 }
16 compte_objet::~compte_objet()
17 {
18     std::cout << "Destruction: il y a maintenant " << --ctr << " objets" << std::endl;
19 }
20 void compte_objet::compte()
21 {
22     std::cout << "Il y a maintenant " << ctr << " objets" << std::endl;
23 }
24
25 void fonction()
26 {
27     compte_objet u, v;
28 }
29
30 int main(int argc, char const *argv[])
31 {
32     // void fonction();
33     compte_objet::compte(); compte_objet a; compte_objet::compte(); fonction(); compte_objet b; compte_objet::compte(); return 0;
34 }
```

Voilà l’affichage après l’exécution de ce programme.12

C:\Users\taffa\OneDrive\Bureau\C++\Achraf-TAFFAH-GLSID1-2022-CHAPITRE3\Cour\exemple9.exe

```
Il y a maintenant 0 objets
Construction: il y a maintenant 1 objets
Il y a maintenant 1 objets
Construction: il y a maintenant 2 objets
Construction: il y a maintenant 3 objets
Destruction: il y a maintenant 2 objets
Destruction: il y a maintenant 1 objets
Construction: il y a maintenant 2 objets
Il y a maintenant 2 objets
Destruction: il y a maintenant 1 objets
Destruction: il y a maintenant 0 objets
```

```
-----
Process exited after 0.1499 seconds with return value 0
Appuyez sur une touche pour continuer...
```

## EXEMPLE 10 :

### LE MOT CLE "THIS"

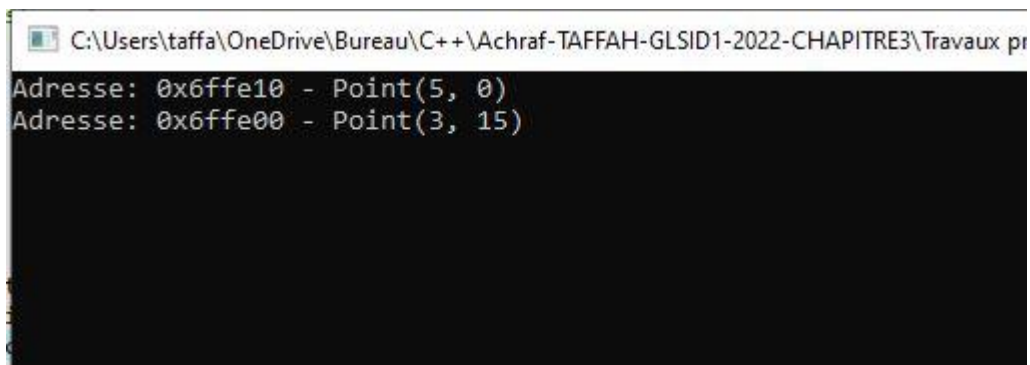
```
example10.cpp
1 #include <iostream>
2
3 class Point
4 {
5 private:
6     int x;
7     int y;
8     std::string color;
9
10 public:
11     void affiche() const;
12 };
13
14 void Point::affiche() const
15 {
16     std::cout << "Point(" << x << ", " << y << ", " << color << ")" << std::endl;
17 }
```

## 2. Exercices :

### EXERCICE 1 & 2 :

```
exercice.cpp
1 #include <iostream>
2
3 class Point
4 {
5 private:
6     int x;
7     int y;
8
9 public:
10     Point(int abs = 0, int ord = 0) : x(abs), y(ord) {}
11     void affiche() const;
12     int coincide(Point *) const;
13 };
14
15 void Point::affiche() const
16 {
17     std::cout << "Adresse: " << this << " - Point(" << x << ", " << y << ")" << std::endl;
18 }
19
20 int Point::coincide(Point *p) const
21 {
22     return (this->x == p->x && this->y == p->y);
23 }
24
25 int main(int argc, char const *argv[])
26 {
27     Point a(5), b(3, 15);
28     a.affiche();
29     b.affiche();
30     std::cin.get();
31     return 0;
32 }
```

Voilà l’affichage après l’exécution de ce programme.140

A screenshot of a Windows command prompt window. The title bar shows the file path: C:\Users\taffa\OneDrive\Bureau\C++\Achraf-TAFFAH-GLSID1-2022-CHAPITRE3\Travaux pr. The command prompt displays two lines of output: 'Adresse: 0x6ffe10 - Point(5, 0)' and 'Adresse: 0x6ffe00 - Point(3, 15)'. The background of the command prompt is black, and the text is white.

## CONCLUSION

ON APPELLE FONCTION UN SOUS-PROGRAMME QUI PERMET D'EFFECTUER UN ENSEMBLE D'INSTRUCTIONS PAR SIMPLE APPEL DE LA FONCTION DANS LE **CORPS** DU PROGRAMME PRINCIPAL. LES FONCTIONS PERMETTENT D'EXECUTER DANS PLUSIEURS PARTIES DU PROGRAMME UNE SERIE D'INSTRUCTIONS, CELA PERMET UNE SIMPLICITE DU CODE ET DONC UNE TAILLE DE PROGRAMME MINIMALE. D'AUTRE PART, UNE FONCTION PEUT FAIRE APPEL A ELLE-MEME, ON PARLE ALORS DE FONCTION RECURSIVE (IL NE FAUT PAS OUBLIER DE METTRE UNE CONDITION DE SORTIE AU RISQUE SINON DE NE PAS POUVOIR ARRETER LE PROGRAMME...).