



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

# COMPTE RENDU DES TRAVAUX PRATIQUES DE LA CHAPITRE 4

Construction et Destruction et initialisation  
des objets

09/04/2022

Préparé par  
ACHRAF TAFFAH

Encadré par  
PR. KHALIFA MANSOURI

# TABLE DES MATIERES

<b>introduction .....</b>	<b>3</b>
<b>Partie pratique .....</b>	<b>4</b>
<b>1. Exemples de cour : .....</b>	<b>4</b>
REMARQUE : .....	4
Exemple : .....	4
Résultats : .....	4
<b>Construction et destruction des objets statiques .....</b>	<b>5</b>
Exemple : .....	5
Résultats : .....	5
<b>Construction et destruction des objets globaux .....</b>	<b>6</b>
Exemple : .....	6
Résultats : .....	6
<b>Construction et destruction des objets temporaires .....</b>	<b>7</b>
Exemple : .....	7
Résultats : .....	7
<b>Construction et destruction des objets dynamiques .....</b>	<b>8</b>
Exemple : .....	8
Résultats : .....	8
<b>Initialisation des objets .....</b>	<b>9</b>
Exemple 1 : .....	9
Résultats : .....	9
Exemple 2 : .....	9
Résultats : .....	10
Exemple 3 : .....	11
Résultats : .....	11
<b>Rôle du constructeur lorsqu'une fonction retourne un objet .....</b>	<b>12</b>
Exemple 1 : .....	12
Résultats : .....	12
Exemple 2 : .....	12
Résultats : .....	13
<b>Les tableaux d'objets .....</b>	<b>14</b>

Exemple :	14
Résultats :	14
<b>1. TP4 :</b>	15
Le fichier header :	15
La definition des methodes existent dans le fichier header	15
le fichier main pour tester la classe et les methodes.....	16
Le resultat apres la compilation de ce programme.....	17
<b>conclusion.....</b>	17

## INTRODUCTION

LE CONSTRUCTEUR EST LA FONCTION MEMBRE APPELEE AUTOMATIQUEMENT LORS DE LA CREATION D'UN OBJET (EN STATIQUE OU EN DYNAMIQUE). CETTE FONCTION MEMBRE EST LA PREMIERE FONCTION MEMBRE A ETRE EXECUTEE, IL S'AGIT DONC D'UNE FONCTION PERMETTANT L'INITIALISATION DES VARIABLES. LE CONSTRUCTEUR D'UN OBJET PORTE LE MEME NOM QUE LA CLASSE ET NE POSSEDE AUCUNE VALEUR DE RETOUR (MEME PAS *VOID*).

# PARTIE PRATIQUE

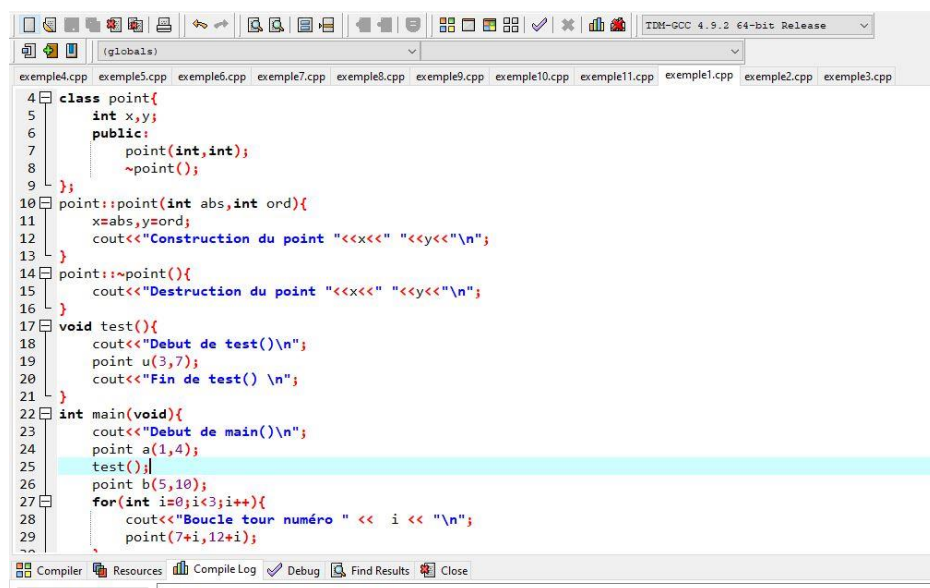
## 1. Exemples de cour :

REMARQUE :

L'objectif de ces exemples est d'étudier soigneusement à quel moment sont créés puis détruits les objets déclarés.

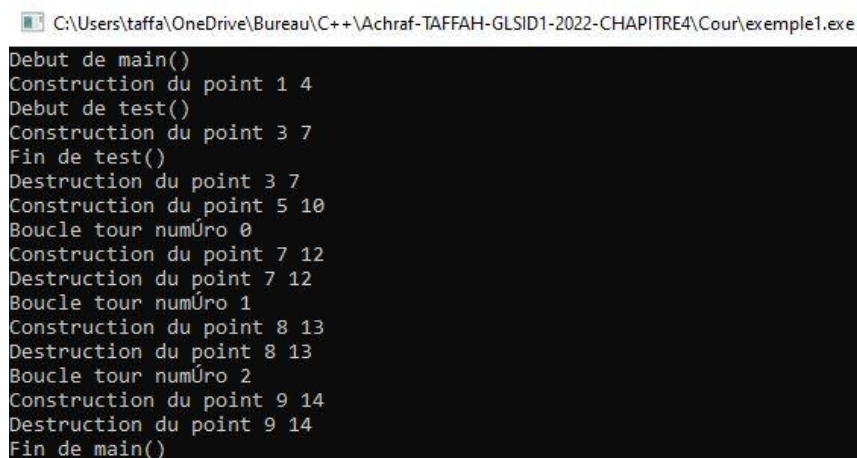
## Construction et destruction des objets automatiques

EXEMPLE :



```
4 class point{
5     int x,y;
6     public:
7         point(int,int);
8         ~point();
9 };
10 point::point(int abs,int ord){
11     x=abs,y=ord;
12     cout<<"Construction du point "<<x<<" "<<y<<"\n";
13 }
14 point::~~point(){
15     cout<<"Destruction du point "<<x<<" "<<y<<"\n";
16 }
17 void test(){
18     cout<<"Debut de test()\n";
19     point u(3,7);
20     cout<<"Fin de test() \n";
21 }
22 int main(void){
23     cout<<"Debut de main()\n";
24     point a(1,4);
25     test();
26     point b(5,10);
27     for(int i=0;i<3;i++){
28         cout<<"Boucle tour numéro " << i << "\n";
29         point(7+i,12+i);
30     }
31 }
```

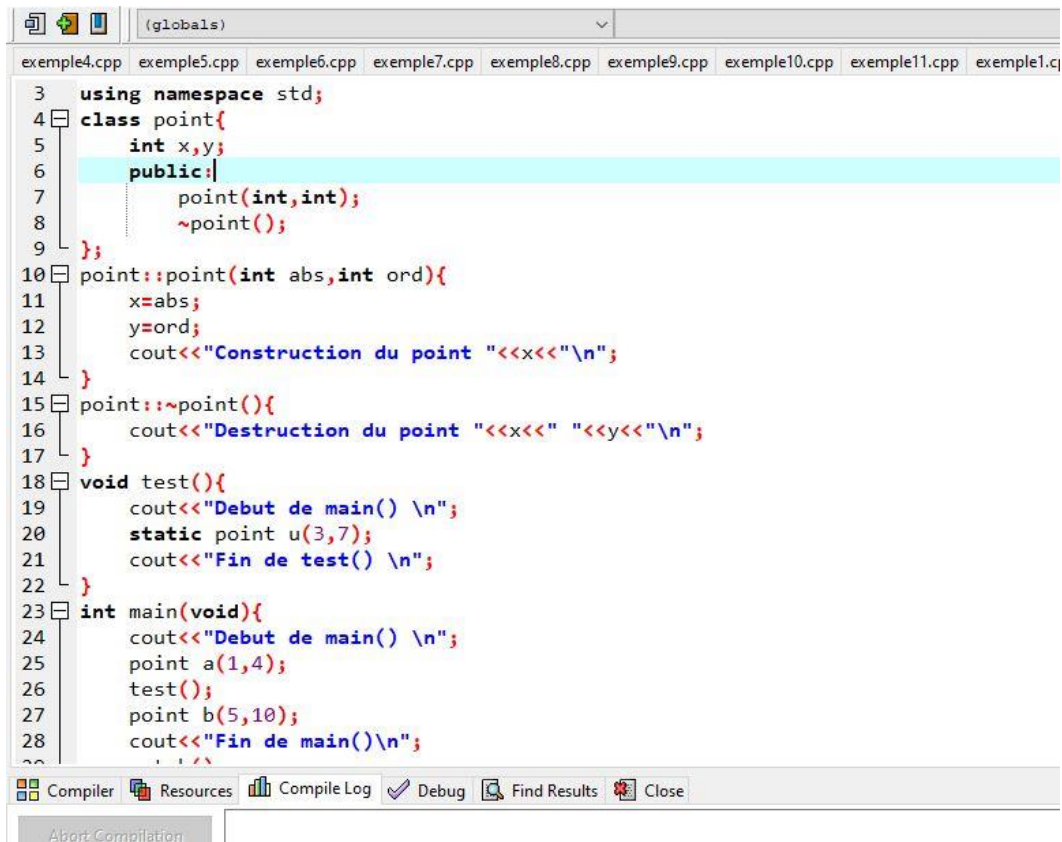
RESULTATS :



```
C:\Users\taffa\OneDrive\Bureau\C++\Achraf-TAFFAH-GLSID1-2022-CHAPITRE4\Cour\exemple1.exe
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 numéro 0
Construction du point 7 12
Destruction du point 7 12
Boucle tour numéro 1
Construction du point 8 13
Destruction du point 8 13
Boucle tour numéro 2
Construction du point 9 14
Destruction du point 9 14
Fin de main()
```

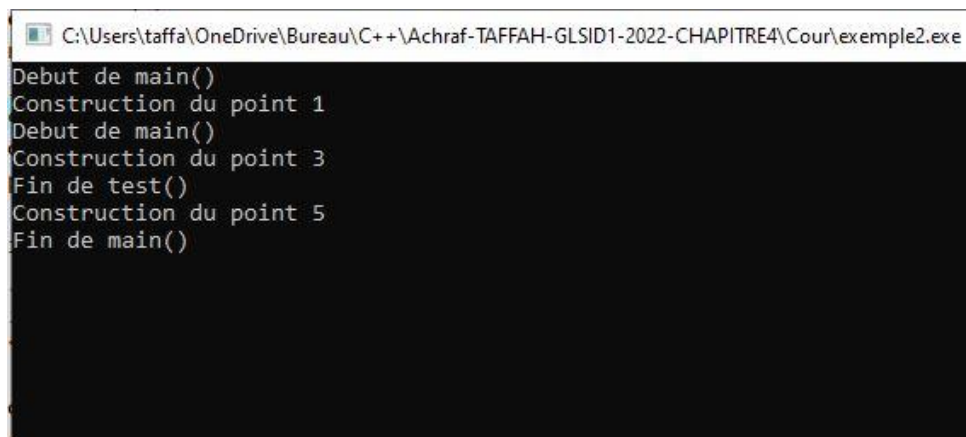
## Construction et destruction des objets statiques

EXEMPLE :



```
3 using namespace std;
4 class point{
5     int x,y;
6     public:
7         point(int,int);
8         ~point();
9 };
10 point::point(int abs,int ord){
11     x=abs;
12     y=ord;
13     cout<<"Construction du point "<<x<<"\n";
14 }
15 point::~~point(){
16     cout<<"Destruction du point "<<x<<" "<<y<<"\n";
17 }
18 void test(){
19     cout<<"Debut de main() \n";
20     static point u(3,7);
21     cout<<"Fin de test() \n";
22 }
23 int main(void){
24     cout<<"Debut de main() \n";
25     point a(1,4);
26     test();
27     point b(5,10);
28     cout<<"Fin de main()\n";
29 }
```

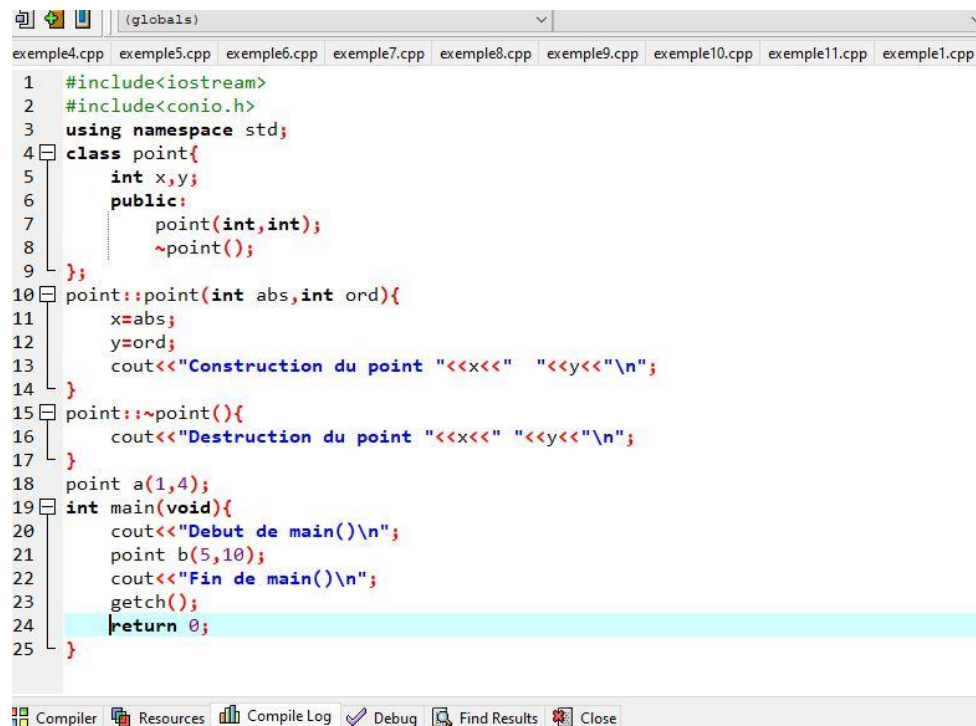
RESULTATS :



```
C:\Users\taffa\OneDrive\Bureau\C++\Achraf-TAFFAH-GLSID1-2022-CHAPITRE4\Cour\exemple2.exe
Debut de main()
Construction du point 1
Debut de main()
Construction du point 3
Fin de test()
Construction du point 5
Fin de main()
```

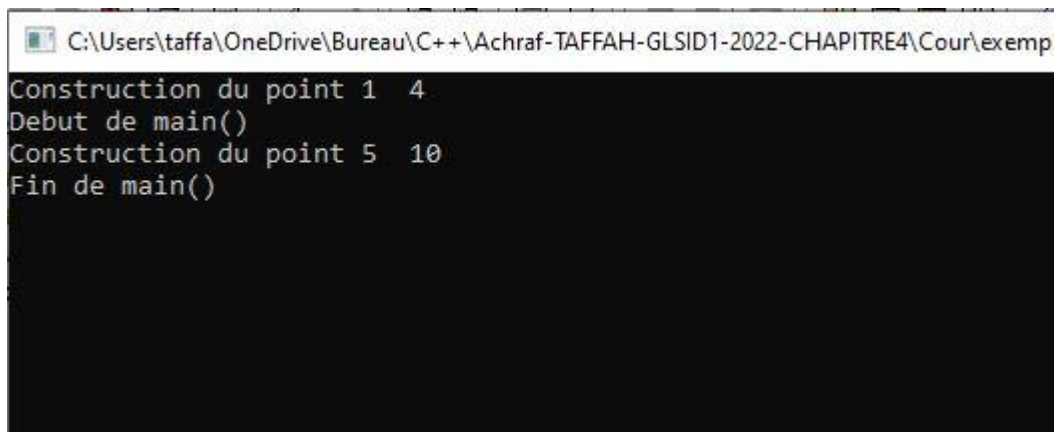
## Construction et destruction des objets globaux

EXEMPLE :



```
1  #include<iostream>
2  #include<conio.h>
3  using namespace std;
4  class point{
5      int x,y;
6      public:
7          point(int,int);
8          ~point();
9  };
10 point::point(int abs,int ord){
11     x=abs;
12     y=ord;
13     cout<<"Construction du point "<<x<<" "<<y<<"\n";
14 }
15 point::~~point(){
16     cout<<"Destruction du point "<<x<<" "<<y<<"\n";
17 }
18 point a(1,4);
19 int main(void){
20     cout<<"Debut de main()\n";
21     point b(5,10);
22     cout<<"Fin de main()\n";
23     getch();
24     return 0;
25 }
```

RESULTATS :



```
C:\Users\taffa\OneDrive\Bureau\C++\Achraf-TAFFAH-GLSID1-2022-CHAPITRE4\Cour\exemp
Construction du point 1  4
Debut de main()
Construction du point 5  10
Fin de main()
```

# Construction et destruction des objets temporaires

EXEMPLE :

```
1 #include<iostream>
2 #include<conio.h>
3 using namespace std;
4 class point {
5     int x,y;
6     public:
7         point(int,int);
8         ~point();
9 };
10 point::point(int abs,int ord){
11     x=abs;
12     y=ord;
13     cout<<"Construction du point "<<x<<" "<<y<<" &l'adresse : " << this << "\n";
14 }
15 point::~~point(){
16     cout<<"Destruction du point "<<x<<" "<<y<<" a l'adresse : " << this << "\n";
17 }
18 int main(void){
19     cout<<"Debut de main () \n";
20     point a(0,0);
21     a=point(1,2);
22     a=point(3,5);
23     cout<<"Fin de main() \n";
24     getch();
25     return 0;
26 }
```

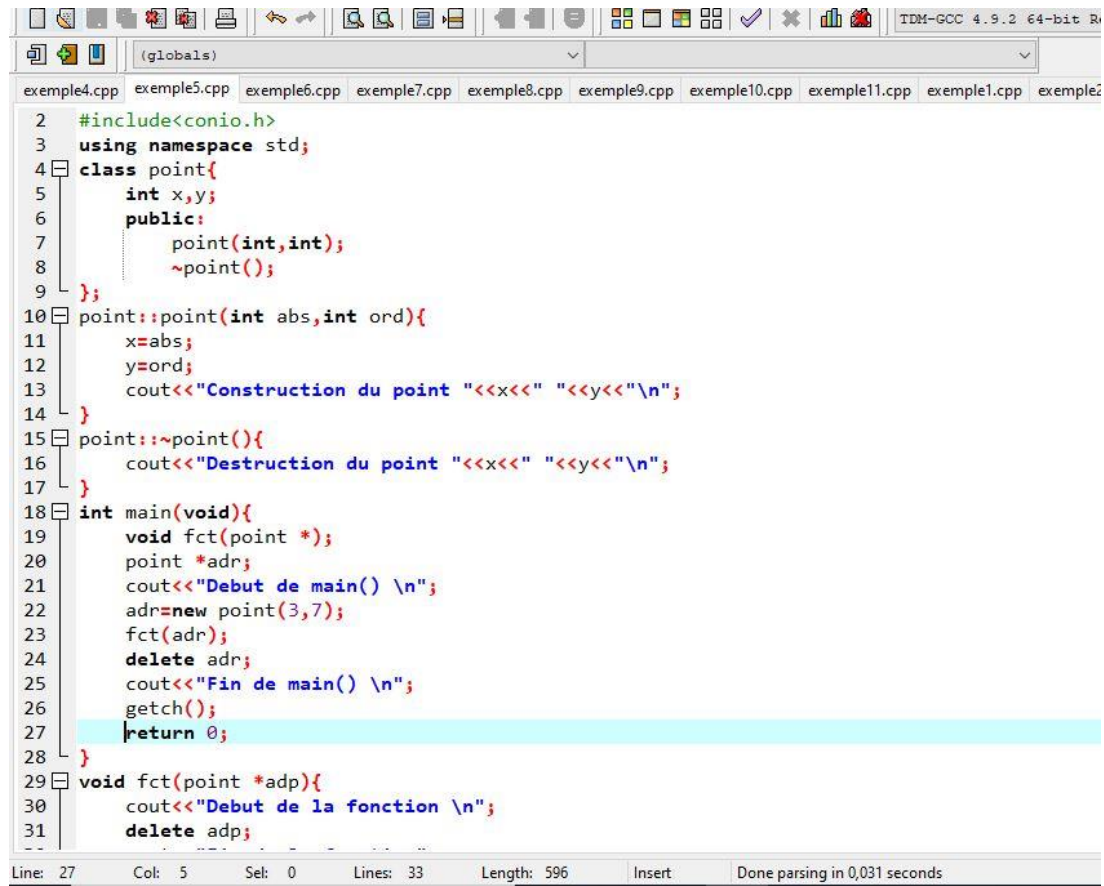
RESULTATS :

```
C:\Users\taffa\OneDrive\Bureau\C++\Achraf-TAFFAH-GLSID1-2022-CHAPITRE4\Cour\exemple4.exe
Debut de main ()
Construction du point 0 0 &l'adresse : 0x6ffde0
Construction du point 1 2 &l'adresse : 0x6ffdf0
Destruction du point 1 2a l'adresse : 0x6ffdf0
Construction du point 3 5 &l'adresse : 0x6ffe00
Destruction du point 3 5a l'adresse : 0x6ffe00
Fin de main()
```



# Construction et destruction des objets dynamiques

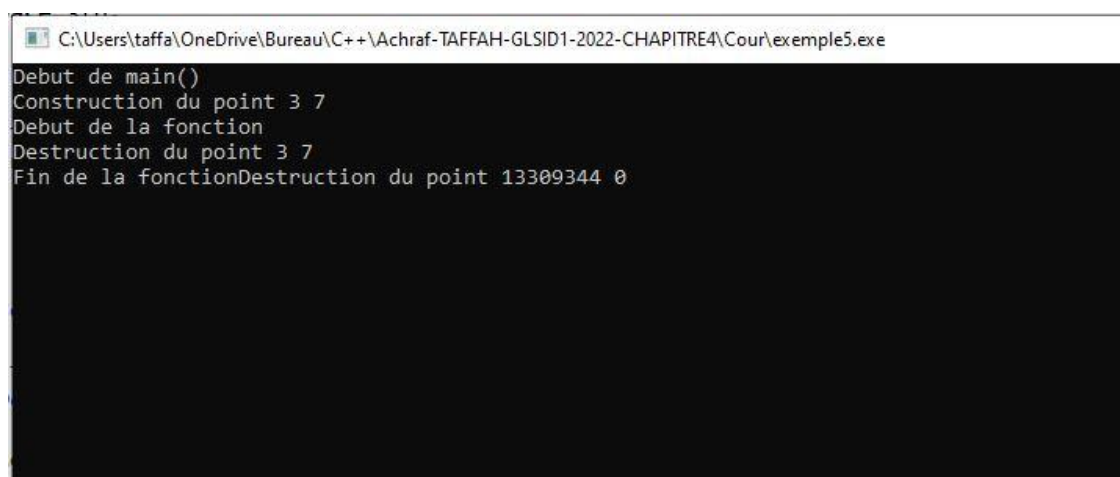
EXEMPLE :



```
2  #include<conio.h>
3  using namespace std;
4  class point{
5      int x,y;
6      public:
7          point(int,int);
8          ~point();
9  };
10 point::point(int abs,int ord){
11     x=abs;
12     y=ord;
13     cout<<"Construction du point "<<x<<" "<<y<<"\n";
14 }
15 point::~~point(){
16     cout<<"Destruction du point "<<x<<" "<<y<<"\n";
17 }
18 int main(void){
19     void fct(point *);
20     point *adr;
21     cout<<"Debut de main() \n";
22     adr=new point(3,7);
23     fct(adr);
24     delete adr;
25     cout<<"Fin de main() \n";
26     getch();
27     return 0;
28 }
29 void fct(point *adp){
30     cout<<"Debut de la fonction \n";
31     delete adp;
```

Line: 27 Col: 5 Sel: 0 Lines: 33 Length: 596 Insert Done parsing in 0,031 seconds

RESULTATS :



```
C:\Users\taffa\OneDrive\Bureau\C++\Achraf-TAFFAH-GLSID1-2022-CHAPITRE4\Cour\exemple5.exe
Debut de main()
Construction du point 3 7
Debut de la fonction
Destruction du point 3 7
Fin de la fonctionDestruction du point 13309344 0
```



## Initialisation des objets

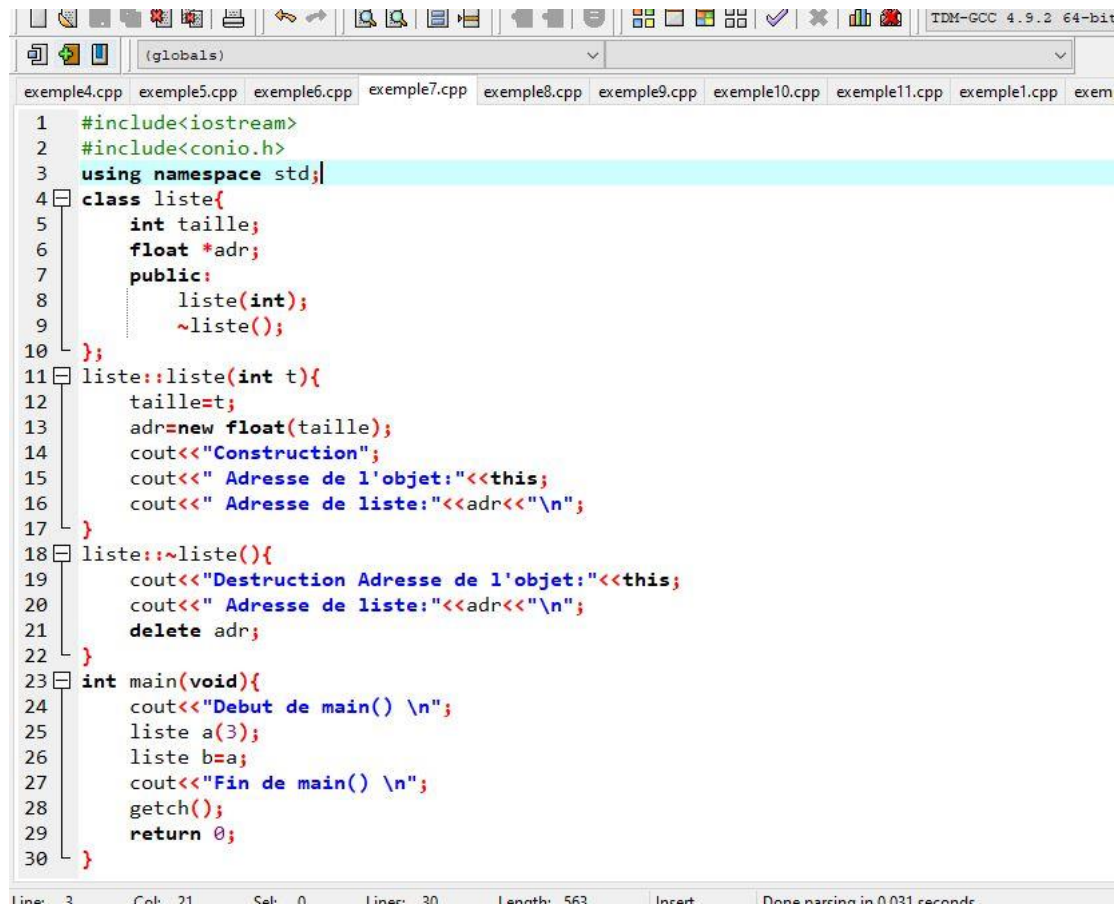
### EXEMPLE 1 :

```
1  #include<iostream>
2  #include<conio.h>
3  #include<dos.h>
4  using namespace std;
5  class point{
6      int x,y;
7      public: point(int,int);
8      ~point();
9  };
10 point::point(int abs,int ord){
11     x=abs;
12     y=ord;
13     cout<<"Construction du point "<<x<<" "<<y<<endl;
14     cout<<" Son adresse : "<<this<<endl;
15 }
16 point::~~point(){
17     cout<<"Destruction du point | "<<x<<" "<<y<<" Son adresse : "<<this<<endl;
18 }
19 int main(void){
20     cout<<"Debut de main() \n";
21     point a(3,7);
22     point b=a;
23     cout<<"Fin de main() \n";
24     getch();
25     return 0;
26 }
```

### RESULTATS :

```
C:\Users\taffa\OneDrive\Bureau\C++\Achraf-TAFFAH-GLSID1-2022-CHAPITRE4\Cour\exemple6.exe
Debut de main()
Construction du point 3 7 Son adresse : 0x6ffe00
Fin de main()
```

### EXEMPLE 2 :



```
1 #include<iostream>
2 #include<conio.h>
3 using namespace std;
4 class liste{
5     int taille;
6     float *adr;
7     public:
8         liste(int);
9         ~liste();
10 };
11 liste::liste(int t){
12     taille=t;
13     adr=new float(taille);
14     cout<<"Construction";
15     cout<<" Adresse de l'objet:"<<this;
16     cout<<" Adresse de liste:"<<adr<<"\n";
17 }
18 liste::~liste(){
19     cout<<"Destruction Adresse de l'objet:"<<this;
20     cout<<" Adresse de liste:"<<adr<<"\n";
21     delete adr;
22 }
23 int main(void){
24     cout<<"Debut de main() \n";
25     liste a(3);
26     liste b=a;
27     cout<<"Fin de main() \n";
28     getch();
29     return 0;
30 }
```

RESULTATS :



```
C:\Users\taffa\OneDrive\Bureau\C++\Achraf-TAFFAH-GLSID1-2022-CHAPITRE4\Cour\exemple7.exe
Debut de main()
Construction Adresse de l'objet:0x6ffe00 Adresse de liste:0x1b18c0
Fin de main()
```

### EXEMPLE 3 :

```

4 class liste{
5     int taille;
6     float *adr;
7     public:
8         liste(int);
9         liste(liste &);
10        ~liste();
11    };
12    liste::liste(int t){
13        taille=t;adr=new float(taille);
14        cout<<"\nConstruction";
15        cout<<" Adresse de l'objet:"<<this;
16        cout<<" Adresse de liste:"<<adr<<"\n";
17    }
18    liste::liste(liste &v){
19        taille=v.taille;
20        adr=new float(taille);
21        for(int i=0;i<taille;i++)
22            adr[i]=v.adr[i];
23        cout<<"\n Constructeur par recopie";
24        cout<<" Adresse de l'objet:"<<this;
25        cout<<" Adresse de liste:"<<adr<<"\n";
26    }
27    liste::~liste(){
28        cout<<"\nDestruction Adresse de l'objet:"<<this;
29        cout<<" Adresse de liste:"<<adr<<"\n";
30        delete adr;
31    }
32    int main(void){
33        cout<<"Debut de main() \n";
34    }

```

### RESULTATS :

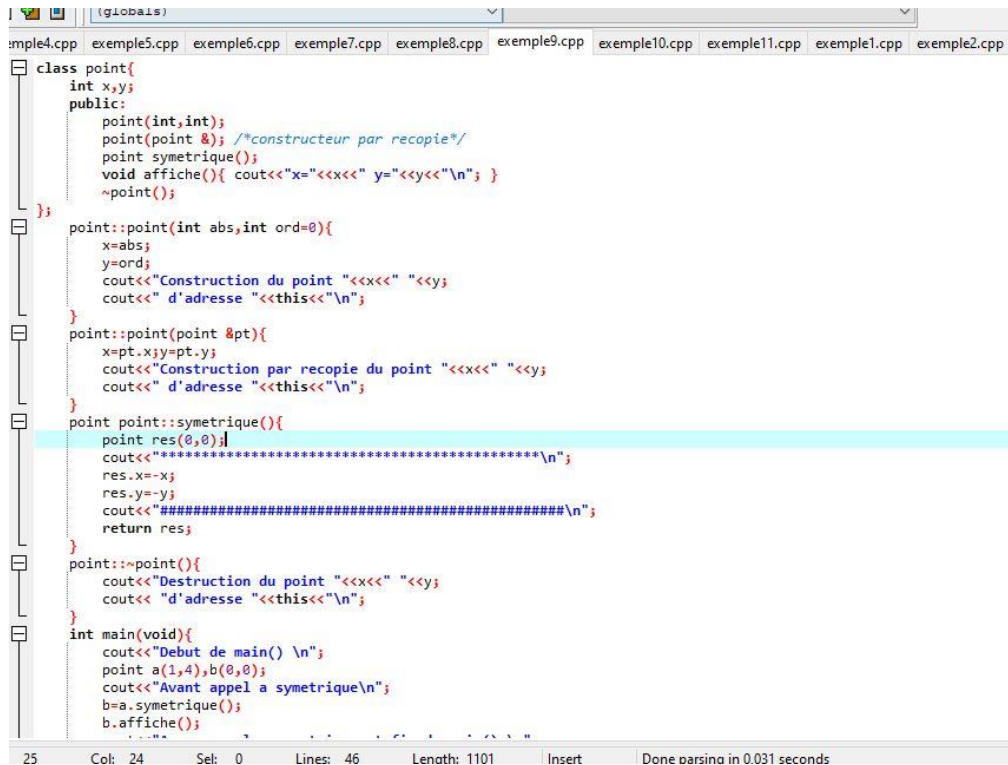
```

C:\Users\taffa\OneDrive\Bureau\C++\Achraf-TAFFAH-GLSID1-2022-CHAPITRE4\Cour\exemple8.exe
Debut de main()
Construction Adresse de l'objet:0x6ffe00 Adresse de liste:0xd118c0
Constructeur par recopie Adresse de l'objet:0x6ffdf0 Adresse de liste:0xd118e0
Fin de main()

```

## Rôle du constructeur lorsqu'une fonction retourne un objet

### EXEMPLE 1 :



```
class point{
    int x,y;
public:
    point(int,int);
    point(point &pt); /*constructeur par recopie*/
    point symetrique();
    void affiche(){ cout<<"x="<<x<<" y="<<y<<"\n"; }
    ~point();
};

point::point(int abs,int ord=0){
    x=abs;
    y=ord;
    cout<<"Construction du point "<<x<<" "<<y;
    cout<<" d'adresse "<<this<<"\n";
}

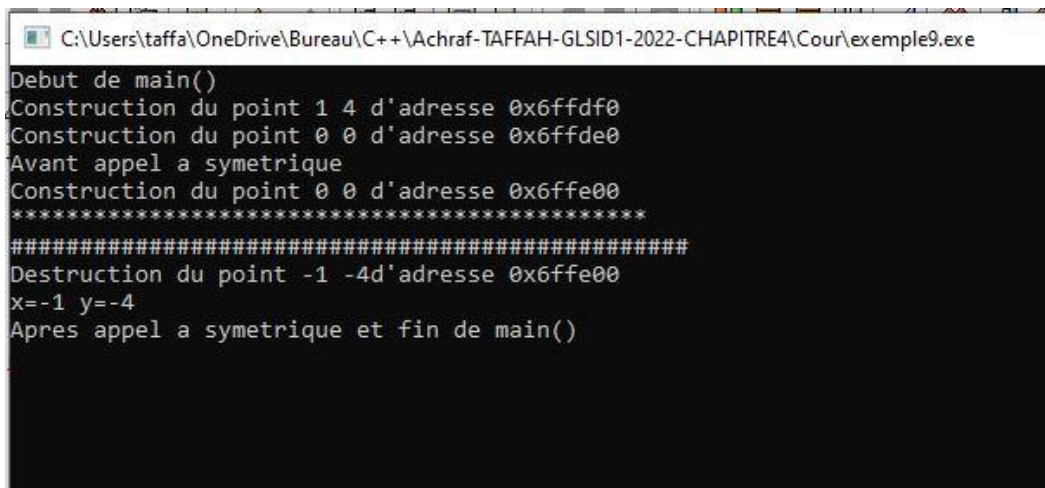
point::point(point &pt){
    x=pt.x;y=pt.y;
    cout<<"Construction par recopie du point "<<x<<" "<<y;
    cout<<" d'adresse "<<this<<"\n";
}

point point::symetrique(){
    point res(0,0);
    cout<<"*****\n";
    res.x=-x;
    res.y=-y;
    cout<<"*****\n";
    return res;
}

point::~~point(){
    cout<<"Destruction du point "<<x<<" "<<y;
    cout<<" d'adresse "<<this<<"\n";
}

int main(void){
    cout<<"Debut de main() \n";
    point a(1,4),b(0,0);
    cout<<"Avant appel a symetrique\n";
    b=a.symetrique();
    b.affiche();
}
```

### RESULTATS :



```
C:\Users\taffa\OneDrive\Bureau\C++\Achraf-TAFFAH-GLSID1-2022-CHAPITRE4\Cour\exemple9.exe
Debut de main()
Construction du point 1 4 d'adresse 0x6ffdf0
Construction du point 0 0 d'adresse 0x6ffde0
Avant appel a symetrique
Construction du point 0 0 d'adresse 0x6ffe00
*****
*****
Destruction du point -1 -4 d'adresse 0x6ffe00
x=-1 y=-4
Après appel a symetrique et fin de main()
```

### EXEMPLE 2 :

```

TDM-GCC 4.9.2 64-bit Release
(global.s)
exemple4.cpp exemple5.cpp exemple6.cpp exemple7.cpp exemple8.cpp exemple9.cpp exemple10.cpp exemple11.cpp exemple1.cpp exemple2.cpp exemple3.cpp

4 class liste{
5     int taille;
6     float *adr;
7     public:
8         liste(int);
9         liste(liste &);
10        void saisie();
11        void affiche();
12        liste oppose();
13        ~liste();
14    };
15    liste::liste(int t){
16        taille=t;adr=new float(taille);
17        cout<<"Construction";
18        cout<<" Adresse de l'objet:"<<this;
19        cout<<" Adresse de liste:"<<adr<<"\n";
20    }
21    liste::liste(liste &v){
22        taille=v.taille;
23        adr=new float[taille];
24        for(int i=0;i<taille;i++) adr[i]=v.adr[i];
25        cout<<"Construction par recopie";
26        cout<<" Adresse de l'objet:"<<this;
27        cout<<" Adresse de liste:"<<adr<<"\n";
28    }
29    liste::~liste(){
30        cout<<"Destruction adresse de l'objet:"<<this;
31        cout<<" Adresse de liste:"<<adr<<"\n";
32        delete adr;
33    }
34    void liste::saisie(){
35        int i;
36        for(i=0;i<taille;i++){
37            cout<<"Entrer un nombre:";
38            cin>>(adr[i]);
39        }
40    }
41    void liste::affiche(){
42        ...
43    }
44    ...
45    ...
46    ...
47    ...
48    ...
49    ...
50    ...
51    ...
52    ...
53    ...
54    ...
55    ...
56    ...
57    ...
58    ...
59    ...
60    ...
61    ...
62    ...
63    ...
64    ...
65    ...
66    ...
67    ...
68    ...
69    ...
70    ...
71    ...
72    ...
73    ...
74    ...
75    ...
76    ...
77    ...
78    ...
79    ...
80    ...
81    ...
82    ...
83    ...
84    ...
85    ...
86    ...
87    ...
88    ...
89    ...
90    ...
91    ...
92    ...
93    ...
94    ...
95    ...
96    ...
97    ...
98    ...
99    ...
100   ...
101   ...
102   ...
103   ...
104   ...
105   ...
106   ...
107   ...
108   ...
109   ...
110   ...
111   ...
112   ...
113   ...
114   ...
115   ...
116   ...
117   ...
118   ...
119   ...
120   ...
121   ...
122   ...
123   ...
124   ...
125   ...
126   ...
127   ...
128   ...
129   ...
130   ...
131   ...
132   ...
133   ...
134   ...
135   ...
136   ...
137   ...
138   ...
139   ...
140   ...
141   ...
142   ...
143   ...
144   ...
145   ...
146   ...
147   ...
148   ...
149   ...
150   ...
151   ...
152   ...
153   ...
154   ...
155   ...
156   ...
157   ...
158   ...
159   ...
160   ...
161   ...
162   ...
163   ...
164   ...
165   ...
166   ...
167   ...
168   ...
169   ...
170   ...
171   ...
172   ...
173   ...
174   ...
175   ...
176   ...
177   ...
178   ...
179   ...
180   ...
181   ...
182   ...
183   ...
184   ...
185   ...
186   ...
187   ...
188   ...
189   ...
190   ...
191   ...
192   ...
193   ...
194   ...
195   ...
196   ...
197   ...
198   ...
199   ...
200   ...
201   ...
202   ...
203   ...
204   ...
205   ...
206   ...
207   ...
208   ...
209   ...
210   ...
211   ...
212   ...
213   ...
214   ...
215   ...
216   ...
217   ...
218   ...
219   ...
220   ...
221   ...
222   ...
223   ...
224   ...
225   ...
226   ...
227   ...
228   ...
229   ...
230   ...
231   ...
232   ...
233   ...
234   ...
235   ...
236   ...
237   ...
238   ...
239   ...
240   ...
241   ...
242   ...
243   ...
244   ...
245   ...
246   ...
247   ...
248   ...
249   ...
250   ...
251   ...
252   ...
253   ...
254   ...
255   ...
256   ...
257   ...
258   ...
259   ...
260   ...
261   ...
262   ...
263   ...
264   ...
265   ...
266   ...
267   ...
268   ...
269   ...
270   ...
271   ...
272   ...
273   ...
274   ...
275   ...
276   ...
277   ...
278   ...
279   ...
280   ...
281   ...
282   ...
283   ...
284   ...
285   ...
286   ...
287   ...
288   ...
289   ...
290   ...
291   ...
292   ...
293   ...
294   ...
295   ...
296   ...
297   ...
298   ...
299   ...
300   ...
301   ...
302   ...
303   ...
304   ...
305   ...
306   ...
307   ...
308   ...
309   ...
310   ...
311   ...
312   ...
313   ...
314   ...
315   ...
316   ...
317   ...
318   ...
319   ...
320   ...
321   ...
322   ...
323   ...
324   ...
325   ...
326   ...
327   ...
328   ...
329   ...
330   ...
331   ...
332   ...
333   ...
334   ...
335   ...
336   ...
337   ...
338   ...
339   ...
340   ...
341   ...
342   ...
343   ...
344   ...
345   ...
346   ...
347   ...
348   ...
349   ...
350   ...
351   ...
352   ...
353   ...
354   ...
355   ...
356   ...
357   ...
358   ...
359   ...
360   ...
361   ...
362   ...
363   ...
364   ...
365   ...
366   ...
367   ...
368   ...
369   ...
370   ...
371   ...
372   ...
373   ...
374   ...
375   ...
376   ...
377   ...
378   ...
379   ...
380   ...
381   ...
382   ...
383   ...
384   ...
385   ...
386   ...
387   ...
388   ...
389   ...
390   ...
391   ...
392   ...
393   ...
394   ...
395   ...
396   ...
397   ...
398   ...
399   ...
400   ...
401   ...
402   ...
403   ...
404   ...
405   ...
406   ...
407   ...
408   ...
409   ...
410   ...
411   ...
412   ...
413   ...
414   ...
415   ...
416   ...
417   ...
418   ...
419   ...
420   ...
421   ...
422   ...
423   ...
424   ...
425   ...
426   ...
427   ...
428   ...
429   ...
430   ...
431   ...
432   ...
433   ...
434   ...
435   ...
436   ...
437   ...
438   ...
439   ...
440   ...
441   ...
442   ...
443   ...
444   ...
445   ...
446   ...
447   ...
448   ...
449   ...
450   ...
451   ...
452   ...
453   ...
454   ...
455   ...
456   ...
457   ...
458   ...
459   ...
460   ...
461   ...
462   ...
463   ...
464   ...
465   ...
466   ...
467   ...
468   ...
469   ...
470   ...
471   ...
472   ...
473   ...
474   ...
475   ...
476   ...
477   ...
478   ...
479   ...
480   ...
481   ...
482   ...
483   ...
484   ...
485   ...
486   ...
487   ...
488   ...
489   ...
490   ...
491   ...
492   ...
493   ...
494   ...
495   ...
496   ...
497   ...
498   ...
499   ...
500   ...
501   ...
502   ...
503   ...
504   ...
505   ...
506   ...
507   ...
508   ...
509   ...
510   ...
511   ...
512   ...
513   ...
514   ...
515   ...
516   ...
517   ...
518   ...
519   ...
520   ...
521   ...
522   ...
523   ...
524   ...
525   ...
526   ...
527   ...
528   ...
529   ...
530   ...
531   ...
532   ...
533   ...
534   ...
535   ...
536   ...
537   ...
538   ...
539   ...
540   ...
541   ...
542   ...
543   ...
544   ...
545   ...
546   ...
547   ...
548   ...
549   ...
550   ...
551   ...
552   ...
553   ...
554   ...
555   ...
556   ...
557   ...
558   ...
559   ...
560   ...
561   ...
562   ...
563   ...
564   ...
565   ...
566   ...
567   ...
568   ...
569   ...
570   ...
571   ...
572   ...
573   ...
574   ...
575   ...
576   ...
577   ...
578   ...
579   ...
580   ...
581   ...
582   ...
583   ...
584   ...
585   ...
586   ...
587   ...
588   ...
589   ...
590   ...
591   ...
592   ...
593   ...
594   ...
595   ...
596   ...
597   ...
598   ...
599   ...
600   ...
601   ...
602   ...
603   ...
604   ...
605   ...
606   ...
607   ...
608   ...
609   ...
610   ...
611   ...
612   ...
613   ...
614   ...
615   ...
616   ...
617   ...
618   ...
619   ...
620   ...
621   ...
622   ...
623   ...
624   ...
625   ...
626   ...
627   ...
628   ...
629   ...
630   ...
631   ...
632   ...
633   ...
634   ...
635   ...
636   ...
637   ...
638   ...
639   ...
640   ...
641   ...
642   ...
643   ...
644   ...
645   ...
646   ...
647   ...
648   ...
649   ...
650   ...
651   ...
652   ...
653   ...
654   ...
655   ...
656   ...
657   ...
658   ...
659   ...
660   ...
661   ...
662   ...
663   ...
664   ...
665   ...
666   ...
667   ...
668   ...
669   ...
670   ...
671   ...
672   ...
673   ...
674   ...
675   ...
676   ...
677   ...
678   ...
679   ...
680   ...
681   ...
682   ...
683   ...
684   ...
685   ...
686   ...
687   ...
688   ...
689   ...
690   ...
691   ...
692   ...
693   ...
694   ...
695   ...
696   ...
697   ...
698   ...
699   ...
700   ...
701   ...
702   ...
703   ...
704   ...
705   ...
706   ...
707   ...
708   ...
709   ...
710   ...
711   ...
712   ...
713   ...
714   ...
715   ...
716   ...
717   ...
718   ...
719   ...
720   ...
721   ...
722   ...
723   ...
724   ...
725   ...
726   ...
727   ...
728   ...
729   ...
730   ...
731   ...
732   ...
733   ...
734   ...
735   ...
736   ...
737   ...
738   ...
739   ...
740   ...
741   ...
742   ...
743   ...
744   ...
745   ...
746   ...
747   ...
748   ...
749   ...
750   ...
751   ...
752   ...
753   ...
754   ...
755   ...
756   ...
757   ...
758   ...
759   ...
760   ...
761   ...
762   ...
763   ...
764   ...
765   ...
766   ...
767   ...
768   ...
769   ...
770   ...
771   ...
772   ...
773   ...
774   ...
775   ...
776   ...
777   ...
778   ...
779   ...
780   ...
781   ...
782   ...
783   ...
784   ...
785   ...
786   ...
787   ...
788   ...
789   ...
790   ...
791   ...
792   ...
793   ...
794   ...
795   ...
796   ...
797   ...
798   ...
799   ...
800   ...
801   ...
802   ...
803   ...
804   ...
805   ...
806   ...
807   ...
808   ...
809   ...
810   ...
811   ...
812   ...
813   ...
814   ...
815   ...
816   ...
817   ...
818   ...
819   ...
820   ...
821   ...
822   ...
823   ...
824   ...
825   ...
826   ...
827   ...
828   ...
829   ...
830   ...
831   ...
832   ...
833   ...
834   ...
835   ...
836   ...
837   ...
838   ...
839   ...
840   ...
841   ...
842   ...
843   ...
844   ...
845   ...
846   ...
847   ...
848   ...
849   ...
850   ...
851   ...
852   ...
853   ...
854   ...
855   ...
856   ...
857   ...
858   ...
859   ...
860   ...
861   ...
862   ...
863   ...
864   ...
865   ...
866   ...
867   ...
868   ...
869   ...
870   ...
871   ...
872   ...
873   ...
874   ...
875   ...
876   ...
877   ...
878   ...
879   ...
880   ...
881   ...
882   ...
883   ...
884   ...
885   ...
886   ...
887   ...
888   ...
889   ...
890   ...
891   ...
892   ...
893   ...
894   ...
895   ...
896   ...
897   ...
898   ...
899   ...
900   ...
901   ...
902   ...
903   ...
904   ...
905   ...
906   ...
907   ...
908   ...
909   ...
910   ...
911   ...
912   ...
913   ...
914   ...
915   ...
916   ...
917   ...
918   ...
919   ...
920   ...
921   ...
922   ...
923   ...
924   ...
925   ...
926   ...
927   ...
928   ...
929   ...
930   ...
931   ...
932   ...
933   ...
934   ...
935   ...
936   ...
937   ...
938   ...
939   ...
940   ...
941   ...
942   ...
943   ...
944   ...
945   ...
946   ...
947   ...
948   ...
949   ...
950   ...
951   ...
952   ...
953   ...
954   ...
955   ...
956   ...
957   ...
958   ...
959   ...
960   ...
961   ...
962   ...
963   ...
964   ...
965   ...
966   ...
967   ...
968   ...
969   ...
970   ...
971   ...
972   ...
973   ...
974   ...
975   ...
976   ...
977   ...
978   ...
979   ...
980   ...
981   ...
982   ...
983   ...
984   ...
985   ...
986   ...
987   ...
988   ...
989   ...
990   ...
991   ...
992   ...
993   ...
994   ...
995   ...
996   ...
997   ...
998   ...
999   ...
1000  ...

```

RESULTATS :

```

C:\Users\taffa\OneDrive\Bureau\C++\Achraf-TAFFAH-GLSID1-2022-CHAPITRE4\Cour\exemple10.exe
Debut de main()
Construction Adresse de l'objet:0x6ffdf0 Adresse de liste:0xc918c0
Construction Adresse de l'objet:0x6ffde0 Adresse de liste:0xc918e0
Entrer un nombre:

```


## Les tableaux d'objets

EXEMPLE :



```
1  #include<iostream>
2  #include<conio.h>
3  using namespace std;
4  class point{
5      int x,y;
6      public:
7          point(int abs=0,int ord=0){
8              x=abs;
9              y=ord;
10             cout<<"Construction point "<<x<<" "<<y<<"\n";
11         }
12     };
13     class poincol{
14         point p;
15         int couleur;
16         public:
17             poincol(int,int,int);
18     };
19     poincol::poincol(int abs,int ord,int coul) :p(abs,ord){
20         couleur=coul;
21         cout<<"Constructeur poincol "<<coul<<"\n";
22     }
23     int main(void){
24         poincol a(1,3,9);
25         return 0;
26     }
```

RESULTATS :



```
C:\Users\taffa\OneDrive\Bureau\C++\Achraf-TAFFAH-GLSID1-2022-CHAPITRE4\Cour\exemple11.exe
Construction point 1 3
Constructeur poincol 9

-----
Process exited after 0.1802 seconds with return value 0
Appuyez sur une touche pour continuer...
```



## 1. TP4 :

### LE FICHIER HEADER :

```
(globals)
main.cpp tp4.cpp tp4.hpp
1  #ifndef _TP4_
2      #define _TP4_
3      static int i;
4  class Note{
5      public:
6          float value;
7          void input();
8          void print();
9          void set(float);
10         float get();
11         string apprecier();
12     };
13     void Harmonise(Note & note);
14     float Moyenne(Note *note,int nbr);
15     void Appreciation(Note *note,int nbr);
16 #endif
```

### LA DEFINITION DES METHODES EXISTENT DANS LE FICHIER HEADER

```
(globals)
main.cpp tp4.cpp tp4.hpp
1  #include"tp4.hpp"
2  void Note::input(){
3      float n(-1);
4      while(n<0||n>20){
5          cout <<"Entrez la note:";
6          cin >> n;
7          cout << endl;
8      }
9      this->set(n);
10 }
11 void Note::print(){
12     cout << this->value << endl;
13 }
14 void Note::set(float v){
15     this->value = v;
16 }
17 float Note::get(){
18     return this->value;
19 }
20 string Note::apprecier(){
21 }
22 void Harmonise(Note & note){
23 }
24 float Moyenne(Note *note,int nbr){
25 }
26 void Appreciation(Note *note,int nbr){
27 }
```

```

globals)
main.cpp tp4.cpp tp4.hpp
20 string Note::apprecier(){
21     if(this->value<=4) return "Mediocre";
22     if(this->value<=8) return "Insuffisant";
23     if(this->value<=11) return "Passable";
24     if(this->value<=16) return "Bien";
25     if(this->value<=19) return "Tres bien";
26     return "Excellent";
27 }
28 void Harmonise(Note & note){
29     if(note.get()<8){
30         note.set(0);
31         return;
32     }
33     note.set(8);
34 }
35 float Moyenne(Note *note,int nbr){
36     float m = 0;
37     for(i=0;i<nbr;i++){
38         m+= (note+i)->get();
39     }
40     return m/nbr;
41 }
42 void Appreciation(Note *note,int nbr){
43     for(i=0;i<nbr;i++){
44         cout << "Etudiant : "<<i+1<< " "<<(note+i)->apprecier()<<endl;
45     }
}
Resources Compile Log Debug Find Results Close

```

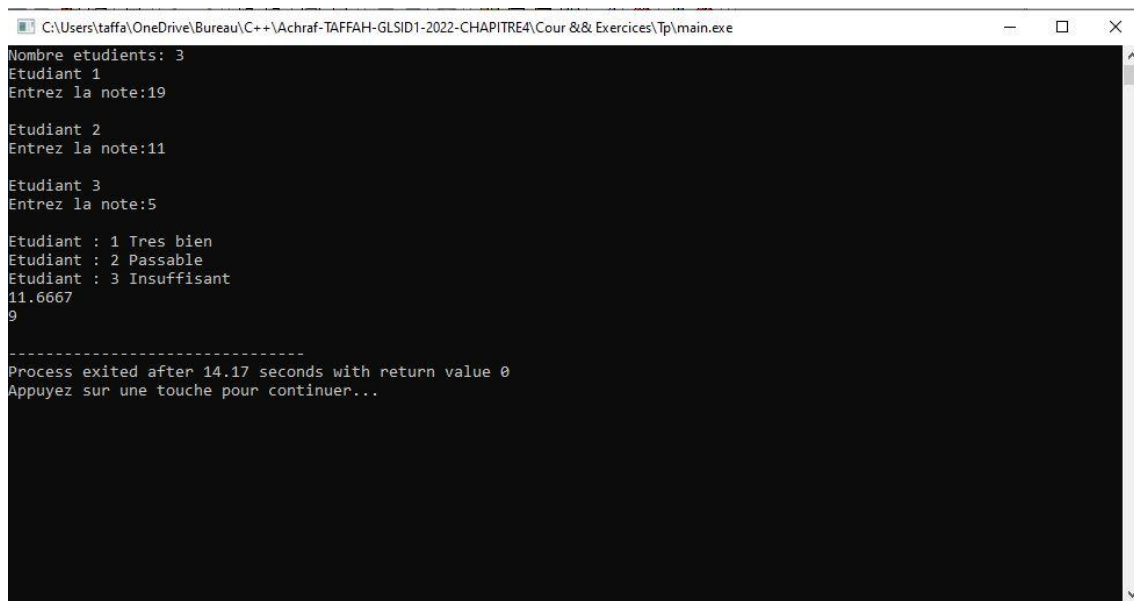
## LE FICHIER MAIN POUR TESTER LA CLASSE ET LES METHODES

```

(globals)
main.cpp tp4.cpp tp4.hpp
1 #include <iostream>
2 using namespace std;
3 #include "tp4.cpp"
4
5 int main(){
6     int nbr_eleves;
7     cout <<"Nombre etudiants: ";
8     cin >> nbr_eleves;
9     Note * notes = new Note[nbr_eleves];
10
11     for(int i=0;i<nbr_eleves;i++){
12         cout << "Etudiant " <<i+1<<endl;
13         (notes+i)->input();
14     }
15
16     Appreciation(notes,nbr_eleves);
17     cout << Moyenne(notes,nbr_eleves)<<endl;
18     for(int i=0;i<nbr_eleves;i++){
19         if((notes+i)->get()<15) Harmonise(*(notes+i));
20     }
21
22     cout << Moyenne(notes,nbr_eleves)<<endl;
23     return 0;
24 }

```

## LE RESULTAT APRES LA COMPILATION DE CE PROGRAMME



```
C:\Users\taffa\OneDrive\Bureau\C++\Achraf-TAFFAH-GLSID1-2022-CHAPITRE4\Cour && Exercices\Tp\main.exe
Nombre etudiants: 3
Etudiant 1
Entrez la note:19

Etudiant 2
Entrez la note:11

Etudiant 3
Entrez la note:5

Etudiant : 1 Tres bien
Etudiant : 2 Passable
Etudiant : 3 Insuffisant
11.6667
9
-----
Process exited after 14.17 seconds with return value 0
Appuyez sur une touche pour continuer...
```

## CONCLUSION

- CONSTRUCTEUR : APPEL AUTOMATIQUE JUSTE APRES LA CREATION DE L'OBJET
- DESTRUCTEUR : APPEL AUTOMATIQUE JUSTE AVANT LA MORT DE L'OBJET
- LE CONSTRUCTEUR ET LE DESTRUCTEUR ASSURENT QUE L'OBJET EST DANS UN ETAT COHERENT.