

ECOLE NORMALE SUPÉRIEURE DE L'ENSEIGNEMENT TECHNIQUE  
DE MOHAMMEDIA

➤ UNIVERSITÉ HASSAN II DE CASABLANCA

## TP2 : programmation orientée objet en c++

ETUDIANT DE FILIER GLSID 1  
ENSET

➤ ENCADRÉ PAR:  
M.K.MANSOURI

➤ REALISER PAR :  
ZAKARIA EL MOURTAZAK

## EXERCICE 1

### LE CODE

```
1  #include <iostream>
2  #include <conio.h>
3  #include <Windows.h>
4  #include <string.h>
5
6  void gotoxy(short a, short b);
7  // les classes
8  class point
9  {
10     int x, y;
11     char couleur[10] = "color ";
12
13 public:
14     void initialiser(int, int, const char *);
15     void deplacer(int, int);
16     void afficher();
17     void effacer();
18 };
19
20 void point::initialiser(int abs, int ord, const char *c)
21 {
22     x = abs;
23     y = ord;
24
25     strcat(couleur, c);
26 }
27
28 void point::deplacer(int dx, int dy)
29 {
30     effacer();
31     x = x + dx;
32     y = y + dy;
33     afficher();
34 }
```

```
36 void point::afficher()
37 {
38     system("cls");
39     system(couleur);
40     gotoxy(x, y);
41     std::cout << "*";
42 }
43
44 void point::effacer()
45 {
46
47     gotoxy(x, y);
48     std::cout << " " << std::endl;
49 }
50
51 void gotoxy(short a, short b)
52 {
53     COORD pos = {a, b};
54     SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), pos);
55 }
56
57 int main(int argc, char const *argv[])
58 {
59     point p1;
60     p1.initialiser(30, 15, "2");
61
62     p1.afficher();
63     getch();
64     p1.deplacer(10, 5);
65     getch();
66
67     return 0;
68 }
```

## L'EXECUTION

\*█

\*█

## EXERCICE 2

### LE CODE

```
void Scene()
{
    point u;
    u.initialiser(30, 15, "2");
    u.afficher();
    getch();
    u.deplacer(10, 5);
    getch();
}
```

```
int main(int argc, char const *argv[])
{
    Scene();
    return 0;
}
```

### L'EXECUTION

\*█

\*█

## EXERCICE 3

### LE CODE

```
point::point(int abs, int ord, const char *c)
{
    x = abs;
    y = ord;

    strcat(couleur, c);
}
```

```
public:
    point(int, int, const char *);
```

### L'EXECUTION

\*█

\*█

## EXERCICE 4

### LE CODE

```
19
20 point::point(int abs, int ord)
21 {
22     x = abs;
23     y = ord;
24     strcat(couleur, "2");
25 }
26
```

```
28 point::~~point()
29
30     std::cout << "\nla dernier point a des coordonnées (x = " << x << " , y = " << y << " )" << std::endl;
31
32
```

## EXERCICE 6

### LE CODE

```
1  #include <iostream>
2  #include <stdlib.h>
3  #include <conio.h>
4  class SuiteArithmetique
5  {
6      int nbval, *val;
7
8  public:
9      SuiteArithmetique(int, int);
10     SuiteArithmetique();
11     ~SuiteArithmetique();
12     void afficher();
13 };
14
15 SuiteArithmetique::SuiteArithmetique(int nb, int mul)
16 {
17     int i;
18     nbval = nb;
19     val = new int[nbval];
20     for (int i = 0; i < nbval; i++)
21         val[i] = i * mul;
22 }
23
24 SuiteArithmetique::~SuiteArithmetique()
25 {
26     delete val;
27 }
28
29 void SuiteArithmetique::afficher()
30 {
31     for (int i = 0; i < nbval; i++)
32         std::cout << val[i] << " ";
33     std::cout << "\n";
34 }
35
36 main()
37 {
38     system("cls");
39     SuiteArithmetique suite1(10, 4);
40     suite1.afficher();
41     getch();
42     SuiteArithmetique suite2(6, 8);
43     suite2.afficher();
44     getch();
45 }
```

### L'EXECUTION

```
0 4 8 12 16 20 24 28 32 36
0 8 16 24 32 40
PS C:\Users\user\Desktop\tpCpp\exercice6> |
```

## EXERCICE 7

### LE CODE

```
1  #include <iostream>
2  #include <time.h>
3
4  class hasard
5  {
6  private:
7      int *val;
8      int nbMax;
9
10 public:
11     hasard(int, int);
12     ~hasard();
13     afficher();
14 };
15
16 hasard::hasard(int nb, int max)
17 {
18     val = new int(nbMax = nb);
19     srand(time(NULL));
20     for (int index = 0; index < nbMax; index++)
21         val[index] = rand() % max + 1;
22 }
23
24 hasard::~~hasard()
25 {
26     delete val;
27 }
28
29 hasard::afficher()
30 {
31     for (int index = 0; index < nbMax; index++)
32         std::cout << val[index] << " ";
33     std::cout << std::endl;
34 }
35
36 main(int argc, char const *argv[])
37 {
38     hasard suite1(4, 10);
39     suite1.afficher();
40     getchar();
41 }
42
```

### L'EXECUTION

```
6 1 4 3
```



## EXERCICE 8

### LE CODE

```
1  #include <iostream>
2  #include <math.h>
3
4  class Complexe
5  {
6  private:
7      double x, y;
8
9  public:
10     Complexe(double, double);
11     Complexe(double);
12     Complexe();
13     void set(double, double);
14     double module();
15     void afficher();
16     double real();
17     double img();
18 };
19
20 Complexe::Complexe(double x, double y)
21 {
22     this->x = x;
23     this->y = y;
24 }
25 Complexe::Complexe(double x)
26 {
27     this->x = x;
28     this->y = 0;
29 }
30 Complexe::Complexe()
31 {
32     this->x = 0;
33     this->y = 0;
34 }
35
36 void Complexe::set(double x, double y)
37 {
38     this->x = x;
39     this->y = y;
40 }
41
42 double Complexe::module()
43 {
44     return sqrt(pow(x, 2) + pow(y, 2));
45 }
46
```

```

41
42 double Complexe::module()
43 {
44     return sqrt(pow(x, 2) + pow(y, 2));
45 }
46
47 void Complexe::afficher()
48 {
49     (y != 0) ? (std::cout << x << " + " << y << "i " << std::endl) : (std::cout << x << std::endl);
50 }
51
52 double Complexe::real()
53 {
54     return x;
55 }
56 double Complexe::img()
57 {
58     return y;
59 }
60
61 main(int argc, char const *argv[])
62 {
63     Complexe cm1(1, 2);
64     std::cout << "real = " << cm1.real() << std::endl;
65     std::cout << "imag = " << cm1.img() << std::endl;
66     std::cout << "module = " << cm1.module() << std::endl;
67     cm1.afficher();
68     getchar();
69
70     Complexe cm2(2);
71     std::cout << "real = " << cm2.real() << std::endl;
72     std::cout << "imag = " << cm2.img() << std::endl;
73     std::cout << "module = " << cm2.module() << std::endl;
74     cm2.afficher();
75     getchar();
76
77     Complexe cm3;
78     std::cout << "real = " << cm3.real() << std::endl;
79     std::cout << "imag = " << cm3.img() << std::endl;
80     std::cout << "module = " << cm3.module() << std::endl;
81     cm3.afficher();
82     getchar();
83 }
84

```

## L'EXECUTION

---

```
real = 1  
imag = 2  
module = 2.23607  
1 + 2i
```

```
real = 2  
imag = 0  
module = 2  
2
```

```
real = 0  
imag = 0  
module = 0  
0  
█
```