# 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>
    #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;
        char couleur[10] = "color ";
11
12
13
    public:
       void initialiser(int, int, const char *);
14
        void deplacer(int, int);
15
16
        void afficher();
     void effacer();
17
18
19
    void point::initialiser(int abs, int ord, const char *c)
20
21
22
        x = abs;
23
      y = ord;
24
25
        strcat(couleur, c);
26
27
28
    void point::deplacer(int dx, int dy)
29
        effacer();
30
31
        x = x + dx;
        y = y + dy;
32
        afficher();
33
34
```

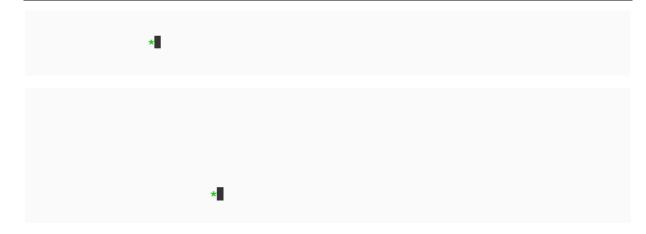
#### UNIVERSITÉ HASSAN II DE CASABLANCA





المدرسة العليا لأساتذة التعليم التقني المحمدية جامعة الحسن الثاني بالدار البيضاء

```
void point::afficher()
37
         system("cls");
38
39
         system(couleur);
        gotoxy(x, y);
std::cout << "*";</pre>
40
41
42
43
44
     void point::effacer()
45
46
         gotoxy(x, y);
47
         std::cout << " " << std::endl;
48
49
50
51
    void gotoxy(short a, short b)
52
        COORD pos = {a, b};
53
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
```



UNIVERSITÉ HASSAN II DE CASABLANCA



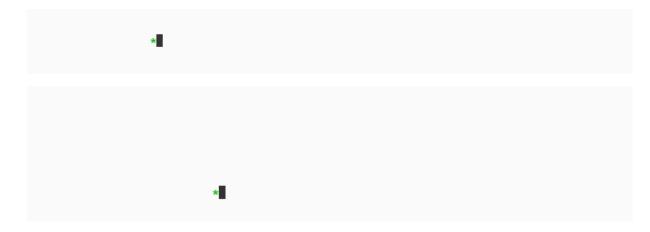


المدرسة العليا لأساتذة التعليم التقني المحمدية جامعة الحسن الثاني بالدار البيضاء

## **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;
```







## **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**

```
T2
    point::point(int abs, int ord)
20
21
22
        x = abs;
23
        y = ord;
        strcat(couleur, "2");
24
25
26
```





## L'EXECUTION



## **EXERCICE 5**

# **LE CODE**

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







## **EXERCICE 6**

#### **LE CODE**

```
#include <iostream>
   #include <stdlib.h>
 3 #include <conio.h>
   class SuiteArithmetique
 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++)
       val[i] = i * mul;
21
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
    main()
36
37
        system("cls");
38
        SuiteArithmetique suite1(10, 4);
      suite1.afficher();
39
40
        getch();
41
        SuiteArithmetique suite2(6, 8);
42
        suite2.afficher();
43
        getch();
44
```

```
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**

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

## **L'EXECUTION**

6143





## **EXERCICE 8**

## **LE CODE**

```
#include <iostream>
1
    #include <math.h>
 3
 4 class Complexe
 5
 6
    private:
 7
     double x, y;
8
9
    public:
10
      Complexe(double, double);
11
        Complexe(double);
        Complexe();
12
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 {
    this->x = x;
27
28
      this->y = 0;
29
30
    Complexe::Complexe()
31
32
       this->x = 0;
33
       this->y = 0;
34
35
    void Complexe::set(double x, double y)
36
37
38
       this->x = x;
39
       this->y = y;
40
41
42
    double Complexe::module()
43
444
      return sqrt(pow(x, 2) + pow(y, 2));
45
46
```

#### UNIVERSITÉ HASSAN II DE CASABLANCA





المدرسة العليا لأساتذة التعليم التقني المحمدية جامعة الحسن الثاني بالدار البيضاء

```
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);</pre>
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);
        std::cout << "real = " << cml.real() << std::endl;
std::cout << "imag = " << cml.img() << std::endl;
64
65
       std::cout << "module = " << cml.module() << std::endl;
66
67
        cml.afficher();
68
      getchar();
69
70
      Complexe cm2(2);
71
         std::cout << "real = " << cm2.real() << std::endl;
        std::cout << "imag = " << cm2.img() << std::endl;
72
        std::cout << "module = " << cm2.module() << std::endl;
73
74
        cm2.afficher();
75
      getchar();
76
       Complexe cm3;
std::cout << "real = " << cm3.real() << std::endl;</pre>
77
78
         std::cout << "imag = " << cm3.img() << std::endl;
79
         std::cout << "module = " << cm3.module() << std::endl;
80
81
         cm3.afficher();
82
         getchar();
83 }
```





```
real = 1
imag = 2
module = 2.23607
1 + 2i
real = 2
imag = 0
module = 2
real = 0
imag = 0
module = 0
Θ
```