

```
1  /*
2  -----
3  Laboratoire : 01 - Class
4  Fichier    : main.cpp
5  Auteur(s)  : Frédéric Korradi et Adrien Barth
6  Date       : 22.02.2017
7
8  But        : Corps principal du programme.
9              Test toutes les fonctions membres des classes Rectangle,
10             Triangle, Carre, Cercle.
11
12 Remarque(s) : ---
13
14 Compilateur : MinGW-g++ 4.8.1
15 -----
16 */
17
18 #include "Carre.h"
19 #include "Cercle.h"
20 #include "Rectangle.h"
21 #include "Triangle.h"
22 #include <iostream>
23
24 using namespace std;
25
26 int main() {
27     // Constructeur par défaut + setter
28     cout << "Constructeur par default + setter : " << endl;
29     Carre carre1;
30     Cercle cercle1;
31     Rectangle rectangle1;
32     Triangle triangle1;
33
34     carre1.setCote(3.14);
35     carre1.details();
36     cout << "Surface: " << carre1.surface() << endl << endl;
37
38     cercle1.setRayon(1.618);
39     cercle1.details();
40     cout << "Surface: " << cercle1.surface() << endl << endl;
41
42     rectangle1.setLargeur(28);
43     rectangle1.setLongueur(42);
44     rectangle1.details();
45     cout << "Surface: " << rectangle1.surface() << endl << endl;
46
47     triangle1.setBase(13);
48     triangle1.setHauteur(15);
49     triangle1.details();
50     cout << "Surface: " << triangle1.surface() << endl << endl;
51
52     // Constructeur surcharge
53     cout << "Constructeur surcharge : " << endl;
54     Carre carre2(34);
55     Cercle cercle2(1.618);
56     Rectangle rectangle2(3, 4);
57     Triangle triangle2(-2, -5);
58
59     carre2.details();
60     cout << "Surface: " << carre2.surface() << endl << endl;
61
62     cercle2.details();
63     cout << "Surface: " << cercle2.surface() << endl << endl;
64
65     rectangle2.details();
66     cout << "Surface: " << rectangle2.surface() << endl << endl;
```

```
67
68     triangle2.details();
69     cout << "Surface: " << triangle2.surface() << endl << endl;
70
71     //Constructeur par défaut sans setter (valeurs a 0)
72     cout << "Constructeur par défaut sans setter (valeurs a 0) :" << endl;
73     Carre carre3;
74     Cercle cercle3;
75     Rectangle rectangle3;
76     Triangle triangle3;
77
78     carre3.details();
79     cout << "Surface: " << carre3.surface() << endl << endl;
80
81     cercle3.details();
82     cout << "Surface: " << cercle3.surface() << endl << endl;
83
84     rectangle3.details();
85     cout << "Surface: " << rectangle3.surface() << endl << endl;
86
87     triangle3.details();
88     cout << "Surface: " << triangle3.surface() << endl << endl;
89
90     return EXIT_SUCCESS;
91 }
```

```
1  /*
2  -----
3  Laboratoire : 01 - Class
4  Fichier     : Carre.h
5  Auteur(s)   : Frédéric Korradi et Adrien Barth
6  Date        : 22.02.2017
7
8  But          : Mettre a disposition une classe rectangle.
9
10 Remarque(s) : ---
11
12 Compilateur  : MinGW-g++ 4.8.1
13 -----
14 */
15 #ifndef CARRE_H
16 #define CARRE_H
17
18 using namespace std;
19
20 class Carre {
21 public:
22     Carre();
23     Carre(double cote);
24     void setCote(const double);
25     double getCote() const;
26     double surface() const;
27     void details() const;
28
29 private:
30     double cote;
31 };
32
33 #endif
```

```
1  /*
2  -----
3  Laboratoire : 01 - Class
4  Fichier     : Carre.cpp
5  Auteur(s)   : Frédéric Korradi et Adrien Barth
6  Date        : 22.02.2017
7
8  But          : Mettre a disposition les fonctions membres de la classe carre.
9
10 Remarque(s) : ---
11
12 Compilateur  : MinGW-g++ 4.8.1
13 -----
14 */
15 #include "Carre.h"
16 #include <iostream>
17
18 Carre::Carre()
19     : cote(.0)
20 {}
21
22 Carre::Carre(double cote)
23     : cote(cote)
24 {}
25
26 void Carre::setCote(const double cote) {
27     if (cote > 0) {
28         this->cote = cote;
29     }
30 }
31 double Carre::getCote() const {
32     return cote;
33 }
34 double Carre::surface() const {
35     return cote * cote;
36 }
37 void Carre::details() const {
38     cout << "--- CARRE ---" << endl
39         << "Cote: " << getCote() << endl;
40 }
```

```
1  /*
2  -----
3  Laboratoire : 01 - Class
4  Fichier     : Cercle.h
5  Auteur(s)   : Frédéric Korradi et Adrien Barth
6  Date        : 22.02.2017
7
8  But          : Mettre a disposition une classe cercle.
9
10 Remarque(s) : ---
11
12 Compilateur  : MinGW-g++ 4.8.1
13 -----
14 */
15 #ifndef CERCLE_H
16 #define CERCLE_H
17
18 using namespace std;
19
20 class Cercle {
21 public:
22     Cercle();
23     Cercle(double rayon);
24     void setRayon(const double);
25     double getRayon() const;
26     double surface() const;
27     void details() const;
28
29 private:
30     double rayon;
31 };
32
33 #endif
```

```
1  /*
2  -----
3  Laboratoire : 01 - Class
4  Fichier     : Cercle.cpp
5  Auteur(s)  : Frédéric Korradi et Adrien Barth
6  Date       : 22.02.2017
7
8  But        : Mettre a disposition les fonctions membres de la classe Cercle.
9
10 Remarque(s) : ---
11
12 Compilateur : MinGW-g++ 4.8.1
13 -----
14 */
15 #include "Cercle.h"
16 #include <iostream>
17
18 Cercle::Cercle()
19     : rayon(.0)
20 {}
21
22 Cercle::Cercle(double rayon)
23     : rayon(rayon)
24 {}
25
26 void Cercle::setRayon(const double rayon) {
27     if (rayon > 0) {
28         this->rayon = rayon;
29     }
30 }
31 double Cercle::getRayon() const {
32     return rayon;
33 }
34 double Cercle::surface() const {
35     return 3.141592653581 * rayon * rayon;
36 }
37 void Cercle::details() const {
38     cout << "--- CERCLE ---" << endl
39         << "Rayon: " << getRayon() << endl;
40 }
```

```
1  /*
2  -----
3  Laboratoire : 01 - Class
4  Fichier    : Rectangle.h
5  Auteur(s)  : Frédéric Korradi et Adrien Barth
6  Date       : 22.02.2017
7
8  But        : Mettre a disposition un classe rectangle.
9
10 Remarque(s) : ---
11
12 Compilateur : MinGW-g++ 4.8.1
13 -----
14 */
15 #ifndef RECTANGLE_H
16 #define RECTANGLE_H
17
18 using namespace std;
19
20 class Rectangle {
21 public:
22     Rectangle();
23     Rectangle(double largeur, double longueur);
24     void setLargeur(const double);
25     void setLongueur(const double);
26     double getLargeur() const;
27     double getLongueur() const;
28     double surface() const;
29     void details() const;
30
31 private:
32     double largeur;
33     double longueur;
34 };
35
36 #endif
```

```
1  /*
2  -----
3  Laboratoire : 01 - Class
4  Fichier     : Rectangle.cpp
5  Auteur(s)  : Frédéric Korradi et Adrien Barth
6  Date       : 22.02.2017
7
8  But         : Mettre a disposition les fonctions membres de la classe rectangle.
9
10 Remarque(s) : ---
11
12 Compilateur : MinGW-g++ 4.8.1
13 -----
14 */
15 #include "Rectangle.h"
16 #include <iostream>
17
18 Rectangle::Rectangle()
19     : largeur(.0), longueur(.0)
20 {}
21
22 Rectangle::Rectangle(double largeur, double longueur)
23     : largeur(largeur), longueur(longueur)
24 {}
25
26 void Rectangle::setLargeur(const double largeur) {
27     if (largeur > 0) {
28         this->largeur = largeur;
29     }
30 }
31 void Rectangle::setLongueur(const double longueur) {
32     if (longueur > 0) {
33         this->longueur = longueur;
34     }
35 }
36 double Rectangle::getLargeur() const {
37     return largeur;
38 }
39 double Rectangle::getLongueur() const {
40     return longueur;
41 }
42
43 double Rectangle::surface() const {
44     return longueur * largeur;
45 }
46 void Rectangle::details() const {
47     cout << "--- RECTANGLE ---" << endl
48         << "Largeur: " << getLargeur() << endl
49         << "Longueur: " << getLongueur() << endl;
50 }
```



```
1  /*
2  -----
3  Laboratoire : 01 - Class
4  Fichier     : Triangle.h
5  Auteur(s)   : Frédéric Korradi et Adrien Barth
6  Date        : 22.02.2017
7
8  But          : Mettre a disposition une classe triangle.
9
10 Remarque(s) : ---
11
12 Compilateur  : MinGW-g++ 4.8.1
13 -----
14 */
15 #ifndef TRIANGLE_H
16 #define TRIANGLE_H
17
18 using namespace std;
19
20 class Triangle {
21 public:
22     Triangle();
23     Triangle(double base, double hauteur);
24     void setBase(const double);
25     void setHauteur(const double);
26     double getBase() const;
27     double getHauteur() const;
28     double surface() const;
29     void details() const;
30
31 private:
32     double base;
33     double hauteur;
34 };
35
36 #endif
```

```
1  /*
2  -----
3  Laboratoire : 01 - Class
4  Fichier      : Rectangle.cpp
5  Auteur(s)    : Frédéric Korradi et Adrien Barth
6  Date         : 22.02.2017
7
8  But          : Mettre a disposition les fonctions membres de la classe Triangle.
9
10 Remarque(s) : ---
11
12 Compilateur  : MinGW-g++ 4.8.1
13 -----
14 */
15 #include "Triangle.h"
16 #include <iostream>
17
18 Triangle::Triangle()
19     : base(.0), hauteur(.0)
20 {}
21
22 Triangle::Triangle(double base, double hauteur)
23     : base(base), hauteur(hauteur)
24 {}
25
26 void Triangle::setBase(const double base) {
27     if (base > 0) {
28         this->base = base;
29     }
30 }
31 void Triangle::setHauteur(const double hauteur) {
32     if (hauteur > 0) {
33         this->hauteur = hauteur;
34     }
35 }
36 double Triangle::getBase() const {
37     return base;
38 }
39 double Triangle::getHauteur() const {
40     return hauteur;
41 }
42 double Triangle::surface() const {
43     return (base * hauteur)/2;
44 }
45 void Triangle::details() const {
46     cout << "--- TRIANGLE ---" << endl
47         << "Base: " << getBase() << endl
48         << "Hauteur: " << getHauteur() << endl;
49 }
```