

## Lab 3

---

### Exercises

#### Question 1.

3 points

Pouvez-vous adapter la classe utilisée dans le programme pour obtenir un résultat similaire à l'exercice suivant :

```
1 #include <iostream>
2
3 int main() {
4     Chose x;
5     std::cout << "Bonjour\n";
6 }
```

Résultat attendu :

```
1 Creation d'un objet de type Chose
2 Bonjour
3 Destruction de l'objet de type Chose
```

Ensuite, quelle serait la sortie du programme suivant, utilisant la même classe "Chose" :

```
1 #include <iostream>
2
3 int main() {
4     Chose *adc = new Chose;
5 }
6
7 main() {
8     int a=10, b=20, c ;
9     c = g(a, b) ;
10    printf ("valeur de g(%d,%d) = %d", a, b, c) ;
11 }
12
13 int g(int x, int y)
14 {
15     return (x*x + 2*x*y + y*y) ;
16 }
```

#### Question 2.

3 points

Quels seront les résultats fournis par l'exécution du programme suivant (ici, la déclaration de la classe demo, sa définition et le programme d'utilisation ont été regroupés en un seul fichier) :

```

1  #include <iostream>
2  using namespace std ;
3  class demo
4  { int x, y ;
5  public :
6      demo (int abs=1, int ord=0) // constructeur I (0, 1 ou 2
          arguments)
7      { x = abs ; y = ord ;
8        cout << "constructeur I : " << x << " " << y << "\n" ;
9      }
10     demo (demo &) ; // constructeur II (par recopie)
11     ~demo () ; // destructeur
12 } ;
13
14 demo::demo (demo & d) // ou demo::demo (const demo & d)
15 { cout << "constructeur II (recopie) : " << d.x << " " << d.y << "
    \n" ;
16   x = d.x ; y = d.y ;
17 }
18
19 demo::~demo ()
20 { cout << "destruction : " << x << " " << y << "\n" ;
21 }
22
23 main ()
24 {
25     void fct (demo, demo *) ; // proto fonction ind[U+FFFD]g n r e fct
26     cout << "debut main\n" ;
27     demo a ;
28     demo b = 2 ;
29     demo c = a ;
30     demo * adr = new demo (3,3) ;
31     fct (a, adr) ;
32     demo d = demo (4,4) ;
33     c = demo (5,5) ;
34     cout << "fin main\n" ;
35 }
36
37 void fct (demo d, demo * add)
38 { cout << "entree fct\n" ;
39   delete add ;
40   cout << "sortie fct\n" ;
41 }

```

### Question 3.

3 points

D clarez une classe Rectangle avec les attributs longueur et largeur. Ajoutez des constructeurs pour initialiser ces attributs avec diff rentes configurations. Assurez-vous  galement d'avoir un destructeur pour la classe. Testez votre classe en cr ant des objets Rectangle avec diff rents constructeurs, puis v rifiez si le destructeur est appel  correctement.

### Question 4.

3 points

Cr er une classe point ne contenant qu'un constructeur sans arguments, un destructeur et un membre donn e priv  repr sentant un num ro de point (le premier cr   portera le num ro 1, le suivant le num ro 2...). Le constructeur affichera le num ro du point cr   et le destructeur affichera le num ro du

point détruit. Écrire un petit programme d'utilisation créant dynamiquement un tableau de 4 points et le détruisant.

### Question 5.

3 points

- Réaliser une classe nommée `set_int` permettant de manipuler des ensembles de nombres entiers. On devra pouvoir réaliser sur un tel ensemble les opérations classiques suivantes : lui ajouter un nouvel élément, connaître son cardinal (nombre d'éléments), savoir si un entier donné lui appartient. Ici, on conservera les différents éléments de l'ensemble dans un tableau alloué dynamiquement par le constructeur. Un argument (auquel on pourra prévoir une valeur par défaut) lui précisera le nombre maximal d'éléments de l'ensemble.
- Écrire, en outre, un programme (main) utilisant la classe `set_int` pour déterminer le nombre d'entiers différents contenus dans 20 entiers lus en données.
- Que faudrait-il faire pour qu'un objet du type `set_int` puisse être transmis

### Question 6.

3 points

Soit la classe point suivante :

```
1 class point
2 { int x, y ;
3 public :
4     point (int abs=0, int ord=0)
5     {
6         x = abs ; y = ord ;
7     }
8 } ;
```

Écrire une fonction indépendante affiche, amie de la classe point, permettant d'afficher les coordonnées d'un point. On fournira séparément un fichier source contenant la nouvelle déclaration (définition) de point et un fichier source contenant la définition de la fonction affiche. Écrire un petit programme (main) qui crée un point de classe automatique et un point de classe dynamique et qui en affiche les coordonnées.

### Question 7.

3 points

On considère une classe `Personne` représentant des individus en temps de confinement. Cette classe comporte trois membres données :

- Deux membres `nom` et `prenom` de type `string` ;
- Un membre `attestation` de type `bool` qui représente le fait que la personne dispose ou non d'une attestation de sortie.

a) Coder deux constructeurs pour la classe `Personne`, respectivement sans argument et avec deux chaînes de caractères en arguments. Dans le premier cas, les membres `nom` et `prenom` seront initialisés avec la chaîne de caractères vide ; dans le second, les deux arguments serviront à initialiser ces deux membres données. Enfin, dans les deux cas, la valeur du membre `attestation` sera initialisée à `false`.

b) Créer une fonction permettant de mettre la valeur du booléen `attestation` à `true`.

Bon courage !!!