Module/ElÃl'ment : POO-C++
Nom de l'enseignant : Youssef Baddi
Niveau : 1ÃÍme annÃl'e

المدرسة العليا للتكنولوجيا
Ecole Supérieure de Technologie
Sidi Bennour - سيدي بنور

TP 6

Exercises

Question 1. 3 points

Quelles erreurs seront détectées par un compilateur C++ dans ce fichier source qui est accepté par un compilateur C ?

```
1  main() {
2     int a=10, b=20, c ;
3     c = g(a, b) ;
4     printf ("valeur de g(%d,%d) = %d", a, b, c) ;
5  }
6
7  g(int x, int y)
8  {
9     return (x*x + 2*x*y + y*y) ;
10 }
```

Question 2. 3 points

Réaliser une classe point permettant de manipuler un point dĂ'un plan. On prévoira :

— un constructeur recevant en arguments les coordonnées (float) d'un point ;

— une fonction membre deplace effectuant une translation définie par ses deux arguments (float) ;

— une fonction membre affiche se contentant d'afficher les coordonnées cartésiennes du point.

Les coordonnées du point seront des membres donnée privés. On écrira séparément :

— un fichier source constituant la déclaration de la classe ;

— un fichier source correspondant à sa définition.

écrire, par ailleurs, un petit programme d'essai (main) déclarant un point, l'affichant, le déplaçant et l'affichant à nouveau.

Question 3. 1 point

Réaliser une classe point, analogue àă la précédente, mais ne comportant pas de fonction affiche. Pour respecter le principe d'encapsulation des données, prévoir deux fonctions membre publiques (nommées abscisse et ordonnee) fournissant en retour l'abscisse et l'ordonnée d'un point. Adapter le petit programme d'essai précédent pour qu'il fonctionne avec cette nouvelle classe.

Question 4. 1 point

Ajouter à la classe précédente (comportant un constructeur et trois fonctions membre deplace, abscisse et ordonnee) de nouvelles fonctions membre :

— homothetie qui effectue une homothétie dont le rapport est fourni en argumentÃ¿ ;

— rotation qui effectue une rotation dont l'angle est fourni en argumentÃ¿ ;

— rho et theta qui fournissent en retour les coordonnées polaires du point.

Question 5. 1 point

Modifier la classe point précédente, de manière que les données (privées) soient mainte- nant les coordonnées polaires d'un point, et non plus ses coordonnées cartésiennes. On évitera de modifier la déclaration des membres publics, de sorte que l'interface de la classe (ce qui est visible pour l'utilisateur) ne change pas.

Question 6. 1 point

Soit la classe point créée dans l'exercice 2, dont la déclaration était la suivante :

```
1  main() {
2     int a=10, b=20, c ;
3     c = g(a, b) ;
4     printf ("valeur de g(%d,%d) = %d", a, b, c) ;
5  }
6
7  g(int x, int y)
8  {
9     return (x*x + 2*x*y + y*y) ;
10 }
```

Adapter cette classe, de manière que la fonction membre affiche fournisse, en plus des coordonnées du point, le nombre d'objets de type point.

Question 7. 1 point

Réaliser une classe nommée set_char permettant de manipuler des ensembles de caractères. On devra pouvoir réaliser sur un tel ensemble les opérations classiques suivantes : lui ajouter un nouvel élément, connaîĆtre son "cardinal" (nombre d'éléments), savoir si un caractère donné lui appartient.
Ici, on n'effectuera aucune allocation dynamique d'emplacements mémoire. Il faudra donc prévoir, en membre donnée, un tableau de taille fixe.
écrire, en outre, un programme (main) utilisant la classe set_char pour déterminer le nombre de caractères différents contenus dans un mot lu en donnée.

Question 8. 1 point

Modifier la classe set_char précédente, de manière à disposer de ce que l'on nomme un ń itérateur ż sur les différents éléments de l'ensemble. Il s'agit d'un mécanisme permettant d'accéder séquentiellement aux différents éléments. On prévoira trois nouvelles fonctions membre : init, qui initialise le processus d'exploration ; prochain, qui fournit la valeur de l'élément suivant lorsqu'il existe et existe, qui précise s'il existe encore un élément non exploré.
On complétera alors le programme d'utilisation précédent, de manière qu'il affiche les différents caractères contenus dans le mot fourni en donnée.

Question 9. 1 point

Soit une classe vecteur3d définie comme suit :

```
1 loatx,y,z public  vecteur3d (float c1=0.0, float c2=0.0, float
      c3=0.0)  x = c1  y = c2  z = c3  ..... ++codetemizetemize
```

Question 10. 1 point

Comment concevoir le type classe chose de façon que ce petit programme :

```
1 main() chose x
2
3 ;cout « "bonjour"
```

fournisse les résultats suivants :
création objet de type chose
bonjour

destruction objet de type chose

Que fournira alors l'exécution de ce programme (utilisant le mêĆme type chose) :

```
main() chose * adc = new chose
```

Question 11. 1 point

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) :

```
include <iostream>using namespace std

;class demo intx,y

;public
:demo (int abs=1, int ord=0)x=abs y=ord cout « "constructeur I  "
   « x « " " « y « ""

;;:;demo (demo )

;demo ()
~;
;demodemo (demo  d)

:: cout « "constructeur II (recopie)  " « d.x « " " « d.y « ""
:;x=d.x y=d.y demodemo ()

::~ cout«"destruction "«x«""«y«""
:;main () void fct (demo, demo *)



;cout « "debut main"
;demo a
;demo b = 2
;demo c = a
;demo * adr = new demo (3,3)
;// proto fonction independante fctfct (a, adr)


;demo d = demo (4,4)
;c = demo (5,5)
;cout « "fin main"
;void fct (demo d, demo * add) cout « "entreee fct"


;delete add
;cout « "sortie fct"
;
```

Question 12. 1 point

1. Réaliser une classe nommée set_int permettant de manipuler des ensembles de nom- bres 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.

2. 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.

3. Que faudrait-il faire pour qu'un objet du type set_int puisse êĆtre transmis par valeur, soit comme argument d'appel, soit comme valeur de retour d'une fonction ?

Question 13. 1 point

Soit la classe point suivante :

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

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 14. 1 point

Créer deux classes (dont les membres donnée sont privés) :

— l'une, nommée vect, permettant de représenter des vecteurs à 3 composantes de type double ; elle comportera un constructeur et une fonction membre d'affichage ;

— l'autre, nommée matrice, permettant de représenter des matrices carrées de dimen- sion 3 x 3 ; elle comportera un constructeur avec un argument (adresse d'un tableau de 3 x 3 valeurs) qui initialisera la matrice avec les valeurs correspondantes.

Réaliser une fonction indépendante prod permettant de fournir le vecteur correspondant au produit d'une matrice par un vecteur. écrire un petit programme de test. On fournira séparé- ment les deux déclarations de chacune des classes, leurs deux définitions, la définition de prod et le programme de test.

Question 15. 1 point

Soit une classe vecteur3d définie comme suit :

```
1 class vecteur3d float x,y,z
2
3 ;public
4 :vecteur3d (float c1=0.0, float c2=0.0, float c3=0.0) x = c1  y =
     c2  z = c3
5
6 ;;;
7 ;
```

Définir les opérateurs == et != de manière qu'ils permettent de tester la coÃŕncidence ou la non-coÃŕncidence de deux points :

a. en utilisant des fonctions membre ;

b. en utilisant des fonctions amies.

Question 16. 1 point

Soit la classe point suivante :

```
class point int x,y

;public
:point (int abs=0, int ord=0) x = abs  y = ord

;;// .....


;
```

a. La munir d'un opérateur de cast permettant de convertir un point en un entier (correspondant aÌĂ son abscisse).

b. Soient alors ces déclarations :

```
point p
;int n
;void fct (int)
;
```

Que font ces instructions :

n = p ; // instruction 1

fct (p) ; // instruction 2

Question 17. 1 point

Quels résultats fournira le programme suivant :

```
include <iostream>using namespace std

;class point int x,y

;public
:point (int abs, int ord) x = abs  y = ord  operator int() //
   "cast" point -> intcout«"***appelint()pourlepoint"«x«""«y«""


;return x

;main() point a(1,5), b(2,8)

;int n1, n2, n3
;n1 = a + 3  // instruction 1
;cout « "n1 = " « n1 « ""
;n2 = a + b  // instruction 2
;cout « "n2 = " « n2 « ""

;double z1, z2
;z1 = a + 3  // instruction 3
;cout « "z1 = "« z1 « ""
;z2 = a + b  // instruction 4
```

```
;cout « "z2 = " « z2 « ""
;
```

Question 18. 1 point

Soient les deux classes suivantes :

```
class B //...public


:B () // constructeur sans argument
;B (int)  // constructeur aÌĂ un argument
;// ...

;class A //...friend operator + (A, A)


;public
:A ()  // constructeur sans argument
;A (int)  // constructeur aÌĂ un argument entier
;A (B)  // constructeur aÌĂ un argument entier de type B
;// ...
;
```

a. Dans un programme contenant les déclarations :

```
A a1, a2, a3
;B b1, b2, b3
;
```

les instructions suivantes seront-elles correctes et, si oui, que feront-elles ?

```
a1 = b1  // instruction 1
;b1 = a1  // instruction 2
;a3=a1+a2 //instruction3
;a3=b1+b2 //instruction4
;b3=a1+a2 //instruction5
;
```

b. Comment obtenir le mêĆme résultat sans définir, dans A, le constructeur A(B) ?

Question 19. 1 point

On dispose d'un fichier nommé point.h contenant la déclaration suivante de la classe point :

```
class point float x,y

;public
:void initialise (float abs=0.0, float ord=0.0) x = abs  y = ord

;;void affiche () cout « "Point de coordonnées  " « x « " " « y «
    ""


;;float abs ()  return x   float ord ()  return y


;
```

— Créer une classe pointb, dérivée de point comportant simplement une nouvelle fonc- tion membre nommée rho, fournissant la valeur du rayon vecteur (premiêĂre coordonnée polaire) d'un point.

— Même question, en supposant que les membres x et y ont été déclarés protégés (protected) dans point, et non plus privés.

— Introduire un constructeur dans la classe pointb.

— Quelles sont les fonctions membre utilisables pour un objet de type pointb ?

Question 20. 1 point

Même question que précédemment, en remplacÌğant simplement l'en-têĆte du constructeur de C par :

```
1 C (int n1=1, int n2=2, int n3=3, float v=0.0)  B(v)
2 :
```

Bon courage ! ! !