

Trimestre Été, 2006

Mohamed Lokbani

IFT1169 – Examen Final –

Inscrivez tout de suite votre nom et le code permanent.

Nom: _____ | Prénom(s): _____ |

Signature: _____ | Code perm: _____ |

Date : mercredi 05 juillet 2006

Durée : 3 heures (de 17h30 à 20h30)

Local : Z-110 ; Pavillon Claire-McNicoll (ancienne aile Z).

Directives:

- Toute documentation permise.
- Calculatrice **non** permise.
- Répondre directement sur le questionnaire.
- Les réponses **doivent être brèves, précises et clairement présentées.**

1. _____ /20 (1.1, 1.2, 1.3)

2. _____ /20 (2.1, 2.2)

3. _____ /20 (3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7)

4. _____ /20 (4.1)

5. _____ /20 (5.1)

Total: _____ /100

Directives officielles

* Interdiction de toute communication verbale pendant l'examen.

* Interdiction de quitter la salle pendant la première heure.

* L'étudiant qui doit s'absenter après la première heure remettra sa carte d'étudiant au surveillant, l'absence ne devant pas dépasser 5 minutes. Un seul étudiant à la fois peut quitter la salle.

* Toute infraction relative à une fraude, un plagiat ou un copiage est signalée par le surveillant au directeur de département ou au professeur qui suspend l'évaluation.

F.A.S

Exercice 1 (20 points)

1.1 Expliquer brièvement pourquoi les opérateurs d'entrée (>>) et de sortie (<<) sont déclarés plutôt comme amis de la classe et non pas comme membres de la classe.

1.2 Une fonction amie d'une classe « X » ne peut accéder qu'aux membres publiques (public) et protégés (protected) de cette classe « X », mais pas aux membres privés. Cette affirmation est-elle vraie ?

1.3 Soit la déclaration suivante:

```
class A {friend class B ; .....};
```

Laquelle des affirmations suivantes est correcte?

- a) La déclaration d'amitié ci-dessus a le même effet que de mettre B comme une sous classe de A, puisque B peut maintenant accéder aux membres protégés de A.
- b) Si B est déjà une sous classe de A, la déclaration d'amitié n'a aucun effet.
- c) La déclaration d'amitié permet à B un meilleur accès aux membres de la classe A que si B était déclarée une sous classe de A.
- d) Aucune des précédentes réponses.

No : _____ Explications : _____

Exercice 2 (20 points)

2.1 Toutes les fonctions génériques doivent être obligatoirement déclarées membres de la classe. Est-ce vrai ?

2.2 Pour chacune des expressions ci-dessous, indiquer si elle est valide ou non et répondre aux questions suivantes :

Si l'expression est valide :

-) La fonction classique ou la fonction générique (template) utilisée (1 2 3).
-) Le type utilisé des différents arguments (promotion ou conversion).
-) Le type utilisé lors de l'instanciation de la fonction générique.

Si l'expression n'est pas valide :

-) Expliquer pourquoi.
-) Lister les différentes figures considérées pour matcher l'appel en question.

```
1) template <class T> T Max(T i, T j){return (i > j) ? i : j;}
2) template <class T> T Max(T i, int j){return (i > j) ? i : j;}
3) int Max (int i, int j){return (i > j) ? i : j;}
double a = 2.0; int b = 2; short c = 2; char d = 2;
```

a) Max (a, a) ;

b) Max (a, b) ;

c) Max (b, a) ;

d) Max (b, b) ;

e) Max (b, d) ;

f) Max (d, d) ;

g) Max (a, c) ;

h) Max (c, c) ;

i) Max (d, a) ;

3.4 Tout en justifiant votre réponse, donner l'instruction qui doit prendre la place de « **Mystere4** » ?

3.5 Tout en justifiant votre réponse, donner l'instruction qui doit prendre la place de « **Mystere5** » ?

3.6 Tout en justifiant votre réponse, donner l'instruction qui doit prendre la place de « **Mystere6** » ?

3.7 Tout en justifiant votre réponse, donner l'instruction qui doit prendre la place de « **Mystere7** » ?

Exercice 4 (20 points) Que va afficher en sortie le programme suivant qui compile et s'exécute correctement.

```
#include <iostream>
#include <vector>
#include <numeric>

using namespace std;

void afficheVecteur(const vector<char>& inVec) {
    cout << "[";
    for (size_t i = 0; i < inVec.size(); i++) {
        cout << inVec[i] << ",";
    }
    cout << "]" << endl;
}

void ninit(char& ref) {
    ref = 'n';
}

class ncherche {
public:
    bool operator()(char ch) { return (ch == 'n'); }
};

int main() {

    vector<char> v(5, 's');
    afficheVecteur(v);
```

```
// Affichage en sortie et explications
```

```
v.push_back('g');
afficheVecteur(v);
```

```
// Affichage en sortie et explications
```

```
for_each(v.begin()+1, v.end()-1, ninit);
afficheVecteur(v);
```

```
// Affichage en sortie et explications
```

<pre>v.push_back('g'); afficheVecteur(v);</pre>
<pre>// Affichage en sortie et explications</pre>
<pre>// Rappel: « accumulate » est une version généralisée de la // sommation. « accumulate(arg1,arg2,arg3) » initialise // le résultat avec la valeur du arg3 et y ajoute la valeur // de chacun des éléments de l'ensemble défini dans // l'intervalle [arg1,arg2). char ch = accumulate(v.begin(), v.begin(), 'a'); cout << ch << endl;</pre>
<pre>// Affichage en sortie et explications</pre>
<pre>afficheVecteur(v);</pre>
<pre>// Affichage en sortie et explications</pre>
<pre>// Rappel: count_if(arg1,arg2,arg3) compte le nombre d'éléments // qui satisfont la condition décrite par (arg3) dans // l'intervalle [arg1, arg2) int i = count_if(v.begin(), v.end(), not1(bind2nd(greater_equal<char>(), 'n'))); cout << i << endl;</pre>
<pre>// Affichage en sortie et explications</pre>

```
vector<char>::iterator loc;

// Rappel : find(arg1,arg2,arg3) retourne le premier itérateur
// i de l'intervalle [arg1,arg2) qui correspond à arg3.
loc = find(v.begin(), v.end(), 'p');
bool b;
if (loc == v.end()) b = true; else b = false;
cout << boolalpha << b << endl;

// Affichage en sortie et explications

ncherche chercheur;
loc = find_if(v.begin(), v.end(), chercheur);
if (loc == v.begin() + 2) b = true; else b = false;
cout << noboolalpha << b << endl;

// Affichage en sortie et explications

vector<char> trouvesmoi;
trouvesmoi.push_back('n');
trouvesmoi.push_back('g');

// Rappel: search(arg1, arg2, arg3, arg4) recherche dans un
// conteneur [arg1,arg2) une séquence de valeurs correspondant
// à la même séquence dans un autre conteneur [arg3,arg4)
loc = search(v.begin(), v.end(),
            trouvesmoi.begin(),trouvesmoi.end());
cout << *loc << endl;

// Affichage en sortie et explications

return 0;
}
```

Exercice 5 (20 points) Soit la classe « IntSet » définie dans le fichier « Q05.h » comme suit :

```
#ifndef _INT_SET_H_
#define _INT_SET_H_

#include <iostream>
#include <vector>
using std::vector;
using namespace std;

// Cette classe hérite de la classe générique vector<int>
// C'est donc une collection ordonnée d'entiers. "vector" est
// optimisé pour permettre d'ajouter des éléments à partir
// de la fin

class IntSet : public vector<int> {
public:

    // Créer un set vide
    IntSet();

    // Initialiser le set à partir d'un tableau d'entiers
    IntSet(int* tableau, int taille);

    // Tester si l'entier "i" est dans le set
    bool est_membre(int i);

    // Ajouter l'entier "i" au set
    void ajout(int i);

    // Retirer l'entier "i" du set
    void efface(int i);

    // Tester si idx est dans le set
    bool operator[](int idx);

    // Cette méthode retourne un nouveau set contenant les 2 sets
    // celui qui a fait l'appel et le set « is » passé en argument
    IntSet set_union(IntSet is);

    // Cette méthode retourne un nouveau set contenant les éléments
    // communs entre le set qui a fait l'appel et le set « is »
    // passé en argument
    IntSet set_intersection(IntSet is);

    // Cette méthode retourne un nouveau set contenant les éléments
    // se trouvant dans le set qui a fait l'appel mais
    // n'apparaissant pas dans le set « is » passé en argument.
    IntSet set_difference(IntSet is);

    // Affichage en sortie
    friend ostream& operator<<(ostream& os, IntSet& is);
};

ostream& operator<<(ostream& os, IntSet& is);

#endif
```

Sachant que les méthodes « `IntSet()` », « `IntSet(int*, int)` », « `bool est_membre(int)` », « `void ajout(int)` », « `void efface(int)` », « `bool operator[](int);` » et « `ostream& operator<<(ostream&, IntSet&)` » ont été déjà définies (i.e. déjà codées donc **fournies**) dans le fichier « `Q05.cpp` ». On vous demande d'écrire dans ce fichier, la définition (i.e. le code complet) des méthodes suivantes :

```
« IntSet set_union(IntSet) »,  
« IntSet set_intersection(IntSet) »  
et « IntSet set_difference(IntSet) ».
```

Bonnes vacances !