

IFT1166 - Session Hiver, Final

Mohamed Lokbani

IFT1166 - FINAL

Inscrivez tout de suite votre nom et code permanent.

Nom: _____ | Prénom(s): _____ |

Signature: _____ | Code perm: _____ |

Date : 19 avril 2005

Durée: 2 heures et 45 minutes (de 18h30 à 21h15)

Local: Z-310, 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.**
- Soyez **BREFS** et **PRÉCIS** dans vos réponses.

1. _____ /15 (1.1, 1.2, 1.3, 1.4)

2. _____ /20 (2.1)

3. _____ /15 (3.1)

4. _____ /20 (4.1 a, b ; 4.2 a, b)

5. _____ /30 (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 remet sa carte d'étudiant au surveillant, l'absence ne devant pas dépasser 5 minutes.

* Un seul étudiant à la fois peut quitter la salle.

* Tout plagiat, copiage ou fraude constitue une infraction et l'étudiant qui commet ce délit se voit attribuer la note F par le professeur.

F.A.S

Question 1 (15 points) Soit la classe mammifère, Mamf, suivante:

```
class Mamf{
    public:
        boolean ADesPattes() {
            return true;
        }
};
```

1.1 Si vous devez créer la classe "Chien" qui hérite de Mamf, conceptuellement, est-ce que vous allez:

- a) surcharger la méthode "ADesPattes" ?
- b) redéfinir la méthode "ADesPattes" ?
- c) utiliser la méthode "ADesPattes" telle quelle, par héritage ?

(a)

(b)

(c)

Pourquoi? (Courte explication)

1.2 En tenant compte de votre précédente réponse (1.1), implémenter la classe "Chien" en conséquence.

1.3 Si vous devez créer la classe "Baleine" qui hérite de Mamf, conceptuellement, est-ce que vous allez:

- a) surcharger la méthode "ADesPattes" ?
- b) redéfinir la méthode "ADesPattes" ?
- c) utiliser la méthode "ADesPattes" telle quelle, par héritage ?

(a)

(b)

(c)

Pourquoi? (Courte explication)

1.4 En tenant compte de votre précédente réponse (1.3), implémenter la classe "Baleine" en conséquence.

Question 2 (20 points) Pour chacune des méthodes (incluant le constructeur) de la classe C, indiquer si la compilation des méthodes de la classe C provoque des erreurs (y compris les warnings). Si oui dites pourquoi et proposer une solution, si c'est possible.

```
1  #include<iostream>
2  using namespace std;
3  class C{
4      int x;
5  public:
6
```

```
    C(int y) x(y) {};
```

correcte

incorrecte

Pourquoi:

```
7      void f(const C &a) {
8          x = a.x;
9      }
```

correcte

incorrecte

Pourquoi:

```
10     void g(C& a) const{
11         x = 10;
12     }
```

correcte

incorrecte

Pourquoi:

```
13  \\\\ int h(int& y){
14  \\\\     return y*0.5;
15  \\\\ }
```

correcte

incorrecte

Pourquoi:

```
16  \\\\ int j(const int* y) const{
17  \\\\     *y = 10;
18  \\\\ }
```

correcte

incorrecte

Pourquoi:

```
19  \\\\ int k(const int& y) const{
20  \\\\     y = 10;
21  \\\\ }
```

correcte

incorrecte

Pourquoi:

```
22  \\\\ };
```

Question 3 (15 points) Tout en justifiant votre réponse, que va afficher en sortie le programme suivant, qui compile et s'exécute correctement.

```
1  #include <iostream>
2  using namespace std;
3
4  class Nombre {
5  public:
6      Nombre(int i) {
7          valeur=i;
8      };
9      Nombre operator- () {
10         cout<< "Moins " << valeur << endl;
11         return Nombre(-valeur);
12     };
13     friend ostream& operator<< (ostream &o, Nombre n);
14     int operator* () {
15         return valeur;
16     };
17     bool operator== (Nombre n) {
18         cout<< "Est-ce que " << valeur << " est égal à " << n.valeur << "?" <<endl;
19         return (valeur==n.valeur);
20     }
21 private:
22     int valeur;
23 };
24 ostream& operator<< (ostream &o, Nombre n) {
25     o<<*n;
26     return o;
27 }
28 int main() {
29     Nombre un(1), moins_un(-1);
30     if (-un == moins_un)
31         cout<< -un << " est équivalent à " << moins_un << endl;
32     else
33         cout<<"Comment ça se fait!"<<endl;
34
35     return 0;
36 }
```


Question 4 (20 points) On vous donne la tâche de réaliser un programme permettant de traiter un ensemble de requêtes soumises par les employés de la compagnie « ift1166 inc. ». Si votre programme devait traiter les requêtes reçues en utilisant les contraintes suivantes :

4.1 « premier arrivé, premier servi » :

a) Tout en justifiant votre réponse, quel est le nom du conteneur STL le plus approprié pour traiter ce genre de contrainte?

b) Quelles sont les méthodes de base, associées à ce type de conteneur? Donner un exemple simple d'utilisation de ces méthodes (de base).

4.2 Ordre alphabétique croissant du nom de la personne, de la liste des employés, ayant soumis la requête.

a) Tout en justifiant votre réponse, comment doit-on adapter le conteneur utilisé dans la question (4.1) pour ce genre de contrainte?

b) Tout en justifiant votre réponse, écrire la déclaration d'un objet r une instance de ce type de conteneur.

Question 5 (30 points) Écrire un programme qui réalise les opérations suivantes:

-1- Lit le contenu d'un fichier passé en argument. Ce fichier contient un certain nombre de mots, chacun se trouvant sur une ligne séparée.

-2- Affiche en sortie:

- a) Le nombre de mots répétés plus d'une fois,
- b) Le mot le plus répété,
- c) Le 10e mot lu, extrait de la liste des mots triée dans un ordre alphabétique croissant.

La forme des lettres dans un mot n'est pas un élément discriminant. Un mot écrit en majuscule ou en minuscule, c'est exactement le même mot.

Si par exemple le fichier test.txt contient les données suivantes:
Le soleil se couche en se levant Il fait beau demain il est venu hier se couche mais fait marquant

L'exécution du programme sera comme suit:
exo5.exe test.txt

L'affichage en sortie sera alors:
Nombre de mots répétés: 4 mots Le mot le plus répété: se Le 10e mot = il

Attention, les critères suivants sont des éléments très importants lors de la correction :

- La simplicité de votre code,
- La clarté de sa présentation,
- L'utilisation des STL,
- L'utilisation des algorithmes de la STL,
- L'optimisation algorithmique de votre programme.

A titre d'exemple, l'utilisation uniquement de « boucles for, while etc. » pour résoudre cet exercice, ne va pas vous permettre d'obtenir la note maximale, même si votre programme est « 100% correct » !

