

Trimestre Automne, 2022

Mohamed Lokbani

IFT1169 – Examen Intra –

Inscrivez tout de suite : votre nom et le code permanent.

Nom : _____ | Prénom(s) : _____

Signature : _____ | Login : _____

Date : vendredi 21 octobre 2022

Durée : 2 heures (de 16h30 à 18h30)

Local : Z-350, Pavillon Claire McNicoll

Directives :

- Toute documentation est permise.
- Appareils électroniques **non** permis.
- Répondre directement sur le questionnaire.
- Les réponses **doivent être brèves, précises, claires et nettement présentées.**

1. _____ /20 (1.1)
2. _____ /20 (2.1, 2.2, 2.3, 2.4, 2.5)
3. _____ /20 (3.1)
4. _____ /20 (4.1, 4.2, 4.3, 4.4, 4.5)
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) trouvez la meilleure correspondance pour les descriptions ci-dessous en y associant un nombre de 1 à 12. Le nombre ne peut-être utilisé qu'une seule fois. Nous avons complété une description à titre d'exemple.

[..... **Les classes (11)**] sont un concept étendu de structures de données : comme les structures de données, elles peuvent contenir des [.....] membres, mais elles peuvent également contenir des [.....] membres.

Une classe est comme [.....] pour un objet.

[.....] est une instantiation d'une classe. En termes de variables, une classe serait [.....] et un objet serait [.....].

[.....] d'accès sont des mots clés dans les langages orientés objet qui définissent l'accessibilité des classes, des méthodes et d'autres membres.

Les membres [.....] d'une classe ne sont accessibles que depuis les autres membres de la même classe (ou depuis leurs "amis").

Les membres [.....] sont accessibles depuis les autres membres de la même classe (ou depuis leurs "amis"), mais aussi depuis les membres de leurs classes dérivées.

Les membres [.....] sont accessibles depuis n'importe quel endroit où l'objet est visible.

Les mots sont à l'infinif, sans les articles, au masculin et au singulier.

modèle	1	
public	2	
private	3	
type	4	
donnée	5	
protected	6	
modificateur	7	
variable	8	
fonction	9	
objet	10	
classe	11	x

Exercice 2 (20 points)

2.1 Expliquez 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.

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

2.3 La liaison dynamique est essentielle pour assurer :

Encapsulation	VRAI	FAUX
Polymorphisme	VRAI	FAUX
Héritage	VRAI	FAUX
Marginalisation	VRAI	FAUX

2.4 Soit le fragment de code suivant :

```
class A {  
public:  
    double uneautre(int,double);  
};  
class B : public A {} ;
```

a) « Surdéfinir » la fonction « double uneautre(int,double); » consiste à faire quoi au juste en pratique?

b) « Redéfinir » la fonction « double uneautre(int,double); » consiste à faire quoi au juste en pratique?

2.5 Un programmeur donné a écrit une série de lignes de code représentant un ensemble de classes et de méthodes. Ce programmeur a décidé de les rendre disponibles, mais sans fournir pour cela le code source. Que doit-il fournir le programmeur au juste pour que cet ensemble de classes et de méthodes soit quand même utilisable?

Expliquez brièvement votre réponse :

Exercice 3 (20 points) ce programme compile et s'exécute :

```
01 #include <iostream>
02
03 using namespace std;
04
05 class Exo03 {
06 public:
07     Exo03(){
08         cout << "Constructeur: " << this << "\n";
09     }
10     Exo03(const Exo03& T){
11         cout << "Recopie: " << this << "\n";
12     }
13     Exo03(Exo03&& T){
14         cout << "Constructeur de déplacement: " << this << "\n";
15     }
16     Exo03& operator=(const Exo03& T){
17         cout << "Opérateur d'affectation: " << this << "\n";
18         return (*this);
19     }
20     Exo03& operator=(Exo03&& T){
21         cout << "Opérateur de déplacement: " << this << "\n";
22         return (*this);
23     }
24     ~Exo03(){
25         cout << "Destructeur: " << this << "\n";
26     }
27 };
28
29 Exo03 UneFonction(Exo03 delta){
30     cout << "UneFonction\n";
31     return delta;
32 }
33
34 int main(){
35     Exo03 alpha;
36     Exo03 bravo = alpha;
37     Exo03 charlie;
38     alpha=charlie;
39
40     bravo = UneFonction(alpha);
41
42     return 0;
43 }
```

Tout en développant votre réponse, que va-t-il afficher en sortie? Prenez une valeur fictive pour « this ».

Exercice 4 (20 points) ce programme compile et s'exécute :

```
01 #include <iostream>
02 #include <string>
03
04 using namespace std;
05
06 template <class T>
07 T Exo04(T arg) {
08     cout << "-1-" << endl;
09     return arg;
10 }
11 template <class T>
12 T* Exo04(T *arg) {
13     cout << "-2-" << endl;
14     return arg;
15 }
16 const char* Exo04(const char *arg) {
17     cout << "-3-" << endl;
18     return arg;
19 }
20 string Exo04(string arg) {
21     cout << "-4-" << endl;
22     return arg;
23 }
24 int main() {
25     char alpha = 'a';
26     string beta("beta");
27
28     cout << Exo04(alpha) << endl;
29     cout << Exo04(3.14159) << endl;
30     cout << Exo04(beta) << endl;
31     cout << *(Exo04(&alpha)) << endl;
32     cout << Exo04("Deux!") << endl;
33
34     return 0;
35 }
```

4.1 Tout en justifiant votre réponse, sélectionnez la signature de la fonction « Exo04 » générique ou classe appropriée et donner l'affichage obtenu en sortie, pour la ligne suivante :

Ligne 28 : cout << Exo04(alpha) << endl;

T Exo04(T arg)	T* Exo04(T *arg)	const char* Exo04(const char *arg)	string Exo04(string arg)
----------------	------------------	------------------------------------	--------------------------

4.2 Tout en justifiant votre réponse, sélectionnez la signature de la fonction « Exo04 » générique ou classe appropriée et donner l’affichage obtenu en sortie, pour la ligne suivante :

Ligne 29 : cout << Exo04(3.14159) << endl;

T Exo04(T arg)	T* Exo04(T *arg)	const char* Exo04(const char *arg)	string Exo04(string arg)
----------------	------------------	------------------------------------	--------------------------

4.3 Tout en justifiant votre réponse, sélectionnez la signature de la fonction « Exo04 » générique ou classe appropriée et donner l’affichage obtenu en sortie, pour la ligne suivante :

Ligne 30 : cout << Exo04(beta) << endl;

T Exo04(T arg)	T* Exo04(T *arg)	const char* Exo04(const char *arg)	string Exo04(string arg)
----------------	------------------	------------------------------------	--------------------------

4.4 Tout en justifiant votre réponse, sélectionnez la signature de la fonction « Exo04 » générique ou classe appropriée et donner l’affichage obtenu en sortie, pour la ligne suivante :

Ligne 31 : cout << *(Exo04(&alpha)) << endl;

T Exo04(T arg)	T* Exo04(T *arg)	const char* Exo04(const char *arg)	string Exo04(string arg)
----------------	------------------	------------------------------------	--------------------------

4.5 Tout en justifiant votre réponse, sélectionnez la signature de la fonction « Exo04 » générique ou classe appropriée et donner l’affichage obtenu en sortie, pour la ligne suivante :

Ligne 32 : cout << Exo04("Deux!") << endl;

T Exo04(T arg)	T* Exo04(T *arg)	const char* Exo04(const char *arg)	string Exo04(string arg)
----------------	------------------	------------------------------------	--------------------------

Exercice 5 (20 points) Écrivez un programme qui réalise ce qui suit :

- 1- Ouvrez le fichier « sortiefic.txt » pour y écrire quelques lignes, puis fermez le fichier.
- 2- Ouvrez de nouveau le fichier « sortiefic.txt » pour :
 - a- calculer le nombre de caractères dans chaque ligne y compris le caractère de retour à la ligne,
 - b- incrémenter ce nombre avec le nombre de caractères y compris le caractère de retour d'une nouvelle ligne,
 - c- les opérations a & b sont réalisées pour toutes les lignes du fichier,
 - d- finalement, écrivez dans le même fichier « sortiefic.txt », en plus du texte original, la somme des caractères lus en fonction du nombre de lignes lues à un instant donné et ceci dans un format prédéfini.

Le fichier « sortiefic.txt » va contenir ce qui suit après les opérations « 1 » et « 2.d ».

Fichier « sortiefic.txt » après -1-	Fichier « sortiefic.txt » après 2.d	Explications du calcul de la dernière ligne et affichage des sommes intermédiaires
abcd↵ efg↵ hi↵ j↵	abcd↵ efg↵ hi↵ j↵ [5] [9] [12] [14]	4 caractères + ↵ = 5 caractères 3 caractères + ↵ = 4 + 5 = 9 2 caractères + ↵ = 3 + 9 = 12 1 caractère + ↵ = 2 + 12 = 14

- Le symbole « ↵ » correspond à un retour à ligne et il est indiqué à titre informatif.
- Il sera tenu compte lors de la correction de la clarté et de la compacité de votre programme.

