

Trimestre Automne, 2019

Mohamed Lokbani

## IFT1169 – Examen Intra –

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

Nom : \_\_\_\_\_ | Prénom(s) : \_\_\_\_\_

Signature : \_\_\_\_\_ | Login : \_\_\_\_\_

Date : mardi 15 octobre 2019

Durée : 2 heures (de 16h30 à 18h30)

Local : Z-310, 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, 1.2, 1.3, 1.4, 1.5)

2. \_\_\_\_\_ /20 (2.1)

3. \_\_\_\_\_ /20 (3.1, 3.2)

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)** soient les deux paires de fichiers « A.h/A.cpp » et « B.h/B.cpp » correspondant respectivement aux classes « A » et « B ».

- « alpha » est une des méthodes privées de la classe « A » et « beta » une des méthodes privées de la classe « B ».

- La méthode « alpha » de la classe « A » fait appel à la méthode « beta » de la classe « B ».

- La méthode « main » est définie dans un fichier à part, le fichier « test.cpp ».

**1.1** Quelle(s) information(s) relative(s) à la classe « B », doit-on ajouter dans la paire de fichiers « A » pour que cette dernière (c.-à-d. la paire « A ») puisse compiler correctement?

**1.2** Quel est le type du fichier créé suite à la compilation de la paire « A »?

**1.3** Quelle est l'erreur générée si vous tentez de produire un exécutable à partir de la paire « A » uniquement? Et pourquoi a-t-elle été provoquée?

**1.4** Si vous commettez une erreur typographique et faite en sorte que la méthode « alpha » appelle la méthode « veta » au lieu de « beta », quelle est l'erreur générée lors de la compilation de la paire « A ».

**1.5** Écrire les instructions relatives au compilateur « gcc », permettant de réaliser l'édition de liens des 3 fichiers « test.cpp », « A.cpp » et « B.cpp »; et qui génère en sortie l'exécutable « exo1.exe ».

**Exercice 2 (20 points)** tout en justifiant votre réponse, ajouter les lignes manquantes et/ou modifier certaines lignes pour que le programme suivant puisse compiler et s'exécuter correctement.

```
01 #include <iostream>
02
03 class Uneautre {
04     int elt; // ne bouge pas de sa place
05 public:
06     Uneautre(int i):elt(i) {}
07     void affiche(Idem x) {
08         cout << elt << " " << x.elt << endl;
09     }
10 };
11
12 class Idem{
13     int elt; // ne bouge pas de sa place
14 public:
15     Idem(int i):elt(i) {}
16     void affiche(Uneautre y) {
17         cout << elt << " " << y.elt << endl;
18     }
19 };
20
21 int main() {
22     Idem zz(55);
23     Uneautre ww(66);
24     zz.affiche(ww);
25     ww.affiche(zz);
26     return 0;
27 }
```

L'affichage obtenu en sortie après avoir apporté toutes les corrections nécessaires :

55 66 66 55
----------------



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

```
01 #include <iostream>
02 #include <vector>
03 #include <algorithm>
04 #include <memory>
05
06 using namespace std;
07
08 void affiche(unique_ptr<int> &x) { cout << x.get() << " : " << *x << endl; }
09
10 int main(int argc, char **argv) {
11     vector<unique_ptr<int> > vec;
12
13     vec.push_back(unique_ptr<int>(new int(9)));
14     vec.push_back(unique_ptr<int>(new int(5)));
15     vec.push_back(unique_ptr<int>(new int(7)));
16
17     sort(vec.begin(), vec.end());
18
19     for_each(vec.begin(), vec.end(), &affiche);
20
21     return 0;
22 }
```

L'affichage obtenu en sortie est comme suit :

```
0x5600ec35ee70 : 9
0x5600ec35ee90 : 7
0x5600ec35eeb0 : 5
```

**3.1** Expliquer le fonctionnement du programme.

**3.2** Modifier le précédent programme afin d'obtenir cet affichage :

```
0x5600ec35eeb0 : 5  
0x5600ec35ee90 : 7  
0x5600ec35ee70 : 9
```

**Exercice 4 (20 points)** le programme compile et s'exécute. Tout en développant votre réponse, que va-t-il afficher en sortie?

```
1  #include <iostream>
2
3  using namespace std;
4
5  void fcn1(){
6      size_t v1 = 42;
7      auto f = [v1] { return v1; };
8      v1 = 0;
9      auto j = f();
10     cout << j << endl;
11 }
12 void fcn2(){
13     size_t v1 = 42;
14     auto f2 = [&v1] { return v1; };
15     v1 = 0;
16     auto j = f2();
17     cout << j << endl;
18 }
19 void fcn3(){
20     size_t v1 = 42;
21     auto f = [v1] () mutable { return ++v1; }; // mutable: on a le droit de modifier la variable
22     v1 = 0;
23     auto j = f();
24     cout << j << endl;
25 }
26 void fcn4(){
27     size_t v1 = 42;
28     auto f2 = [&v1] { return ++v1; };
29     v1 = 0;
30     auto j = f2();
31     cout << j << endl;
32 }
33 void fcn5(){
34     size_t v1 = 42;
35     size_t* const p = &v1;
36     auto f = [p]() { return ++*p; };
37     auto j = f();
38     cout << v1 << " " << j << endl;
39     v1 = 0;
40     j = f();
41     cout << v1 << " " << j << endl;
42 }
43 int main(){
44     fcn1();
45     fcn2();
46     fcn3();
47     fcn4();
48     fcn5();
49     return 0;
50 }
```



**Exercice 5 (20 points)** écrire un programme « C++ » qui détecte si un mot fourni en ligne de commande, il fait partie de la liste des mots fournis sur cette même ligne de commande. Voici quelques utilisations du programme et les résultats obtenus.

Commande	Affichage en sortie
<code>./exo5.exe --chercher=pommes peches raisins bananes pommes mangues</code>	<code>pommes est dans la liste.</code>
<code>./exo5.exe peches raisins --chercher=pommes bananes pommes mangues</code>	<code>pommes est dans la liste.</code>
<code>./exo5.exe --chercher=pommes peches raisins bananes mangues pasteques</code>	<code>pommes n'est pas dans la liste.</code>
<code>./exo5.exe peches raisins bananes mangues pasteques pommes</code>	<code>Usage: exo5.exe --chercher=mot mot1 mot2 ...</code>

