

Trimestre Hiver, 2010

Mohamed Lokbani

IFT1169 – Examen Intra –

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

Nom: _____ | Prénom(s): _____ |

Signature: _____ | Code perm: _____ |

Date : mardi 23 février 2009

Durée : 2 heures (de 18h30 à 20h30)

Local : Z-240 ; Pavillon Claire McNicoll

Directives:

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

1. _____ /11 (1.1, 1.2, 1.3, 1.4, 1.5, 1.6)
2. _____ /15 (2.1, 2.2, 2.3, 2.4, 2.5)
3. _____ /20 (3.1, 3.2, 3.3)
4. _____ /24 (4.1, 4.2, 4.3, 4.4, 4.5, 4.6)
5. _____ /30 (5.1, 5.2, 5.3, 5.4, 5.5, 5.6, 5.7, 5.8, 5.9)

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 (11 points) Encerclez la bonne réponse et donnez une courte explication.

1.1 [Vrai | Faux] Le sigle du cours « Programmation avancée en C++ » est « IFT1166 ».

1.2 [Vrai | Faux] Un objet de la classe dérivée contient tous les attributs de la classe de base, même les attributs privés.

1.3 [Vrai | Faux] Si une méthode est fournie dans la classe de base, cette méthode doit être obligatoirement redéfinie dans la classe dérivée.

1.4 [Vrai | Faux] En C++, par défaut si la classe a un constructeur, elle doit avoir aussi un destructeur.

1.5 [Vrai | Faux] Si la classe X hérite de la classe Y, alors par défaut X ne peut plus être héritée par la suite.

1.6 [Vrai | Faux] Le constructeur de copie est invoqué sur un objet quand ce dernier est un argument passé par référence.

Exercice 2 (15 points) Soit les deux programmes -A- et -B- suivants :

	// Programme -A-		// Programme -B-
01	#include <iostream>	01	#include <iostream>
02		02	
03	namespace UN {	03	namespace UN{
04	int var;	04	int var;
05	void fxy();	05	void fxy();
06	}	06	}
07	void UN::fxy() {}	07	void UN::fxy() {}
08		08	
09	using namespace UN;	09	using namespace UN;
10	using namespace std;	10	using namespace std;
11		11	
12	int var;	12	int main() {
13		13	
14	void fxy(int arg=0){	14	void fxy(int a=0);
15	cout << arg;	15	int var=5;
16	}	16	var++;
17		17	fxy();
18	int main(){	18	cout << var << endl;
19	var=5;	19	return 0;
20	var++;	20	}
21	fxy();	21	
22	cout << var;	22	void fxy(int arg=0){
23	return 0;	23	cout << arg << endl;
24	}	24	}

2.1 Tout en justifiant votre réponse, est-ce que le programme -A- compile-t-il correctement?

(a) oui, il compile correctement | (b) non, il y a un problème à la compilation |

Pourquoi ?

2.2 Si le programme -A- compile correctement, que va-t-il afficher en sortie? Dans le cas contraire (le programme -A- ne compile pas correctement), passer à la question (2.3).

2.3 Tout en justifiant votre réponse, est-ce que **le programme -B-** compile-t-il correctement?

(a) oui, il compile correctement | (b) non, il y a un problème à la compilation |

Pourquoi ?

2.4 **Si le programme -B-** compile correctement, que va-t-il afficher en sortie? Dans le cas contraire (le programme -B- ne compile pas correctement), passer à la question (2.5).

2.5 À travers vos différentes réponses aux questions (2.1 à 2.4), est-ce que les deux programmes -A- et -B- sont équivalents? Pourquoi?

Exercice 3 (20 points)

3.1 Écrire une fonction « int_fois_dix » qui multiplie par dix l'entier passé en argument. La fonction retourne le résultat de cette opération, un entier.

3.2 La fonction « tab_fois_dix » accepte 3 arguments : une fonction « fct », le tableau d'entiers « tab » et un entier « n » la taille du tableau « tab ». La fonction « fct » accepte un seul argument du type entier et retourne une valeur entière.

Écrire le code de la fonction « tab_fois_dix » qui applique la fonction « fct » au tableau « tab » de taille « n ».

3.3 Écrire une fonction « main » qui teste l'utilisation de la fonction « tab_fois_dix ».

Exercice 4 (24 points) Que va afficher en sortie le programme suivant qui compile et s'exécute correctement :

```
01 #include <iostream>
02 using namespace std;
03
04 class U {
05 public:
06     virtual void a( ) {cout << endl;}
07     virtual void b( ) {cout << 'b'; a( );}
08     virtual void c( ) {cout << 'c'; b( );}
09     virtual void d( ) {cout << 'd'; c( );}
10     virtual void e( ) {cout << 'e'; d( );}
11     virtual void f( ) {cout << 'f'; e( );}
12 };
13 class V: public U {
14 public:
15     virtual void b( ) {cout << 9; a( );}
16     virtual void e( ) {cout << 8; d( );}
17 };
18 class W: public V {
19 public:
20     virtual void c( ) {cout << 7; b( );}
21     virtual void f( ) {cout << 6; e( );}
22 };
23 class X: public W {
24 public:
25     virtual void b( ) {cout << 5; a( );}
26     virtual void d( ) {cout << 4; c( );}
27 };
28 class Y: public X {
29 public:
30     virtual void c( ) {cout << 3; b( );}
31     virtual void e( ) {cout << 2; d( );}
32 };
33 class Z: public Y {
34 public:
35     virtual void d( ) {cout << 1; c( );}
36     virtual void f( ) {cout << 0; e( );}
37 };
38 int main( ) {
39     U *u = new U;
40     u->f( );
41     U *v = new V;
42     v->f( );
43     U *w = new W;
44     w->f( );
45     U *x = new X;
46     x->f( );
47     U *y = new Y;
48     y->f( );
49     U *z = new Z;
50     z->f( );
51     return 0;
52 }
```

4.1 L'affichage obtenu en sortie suite à l'exécution de la ligne « 40 » (u->f();) (Développer votre réponse)

4.2 L'affichage obtenu en sortie suite à l'exécution de la ligne « 42 » ($v \rightarrow f(\quad);$) (Développer votre réponse)

4.3 L'affichage obtenu en sortie suite à l'exécution de la ligne « 44 » ($w \rightarrow f(\quad);$) (Développer votre réponse)

4.4 L'affichage obtenu en sortie suite à l'exécution de la ligne « 46 » ($x \rightarrow f(\quad);$) (Développer votre réponse)

4.5 L'affichage obtenu en sortie suite à l'exécution de la ligne « 48 » ($y \rightarrow f(\quad);$) (Développer votre réponse)

4.6 L'affichage obtenu en sortie suite à l'exécution de la ligne « 50 » ($z \rightarrow f(\quad);$) (Développer votre réponse)

Exercice 5 (30 points) La classe « Monnaie » est composée d’une série de méthodes et de deux membres privés du type entier : « dollar » et « cent ». La classe « Monnaie » est décrite dans le fichier « Monnaie.h » comme suit :

	// Monnaie.h
01	#ifndef Monnaie_H
02	#define Monnaie_H
03	
04	#include <iostream>
05	using namespace std;
06	
07	class Monnaie
08	{
09	public:
10	Monnaie() : dollar(0), cent(0) { }
11	Monnaie(int d, int c) : dollar(d), cent(c) { }
12	Monnaie(int allc);
13	
14	Monnaie operator+(const Monnaie &) const;
15	Monnaie operator-(const Monnaie &) const;
16	Monnaie operator-() const;
17	bool operator==(const Monnaie &) const;
18	
19	bool operator<=(const Monnaie &) const;
20	
21	int getDollars() const { return dollar; }
22	int getCents() const { return cent; }
23	
24	friend ostream& operator<<(ostream&, const Monnaie &);
25	friend istream& operator>>(istream&, Monnaie &);
26	friend bool operator>(const Monnaie &, const Monnaie &);
27	
28	private:
29	int dollar;
30	int cent;
31	};
32	
33	bool operator<(const Monnaie &, const Monnaie &);
34	bool operator!=(const Monnaie &, const Monnaie &);
35	
36	#endif

5.1 Est-il nécessaire d’inclure dans le fichier les instructions définies dans les lignes « 01 », « 02 », et « 36 »? Pourquoi?

5.2 Quel est la nature des méthodes définies dans les lignes « 33 » et « 34 »?

Pour ce qui suit, le codé demandé sera inclus dans le fichier « Monnaie.cpp » :

5.3 Le constructeur de la ligne 12, « `Monnaie(int allc);` »;

5.4 L'opérateur de la ligne « 14 ».

5.5 L'opérateur de la ligne « 16 ».

5.6 L'opérateur de la ligne « 19 ».

5.7 L'opérateur de la ligne « 24 ».

5.8 L'opérateur de la ligne « 34 ».

5.9 Le fichier « test_monnaie.cpp » contient la fonction « main » permettant de construire différents éléments de la classe « Monnaie » et de les tester à l'aide des méthodes de la classe « Monnaie ». Nous allons supposer pour ce qui suit, que tout le contenu de la classe « Monnaie » a été défini dans le fichier « Monnaie.cpp ».

Écrire un simple programme permettant de tester les éléments définis dans les lignes : « 10 », « 11 », « 12 », « 15 », « 16 », « 17 », « 24 » et « 33 », de la classe « Monnaie ».