

IFT1166 - Session Hiver, Intra

Mohamed Lokbani

**IFT1166 - INTRA**

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

Signature: \_\_\_\_\_ | Code perm: \_\_\_\_\_

Date: 16 Février 2K

Durée: 2 heures (de 18h30 à 20h30) Local: P-310

**Directives:**

- Il vous est permis d'utiliser un livre de votre choix.
- Les documents de cours ne sont pas autorisés.
- Ordinateurs personnels prohibés.
- Calculatrice (simple) permise.

- Répondre directement sur le questionnaire.

1. \_\_\_\_\_ /20

2. \_\_\_\_\_ /10

3. \_\_\_\_\_ /20

4. \_\_\_\_\_ /20

5. \_\_\_\_\_ /30

Total: \_\_\_\_\_ /100

## Question 1 (20 points)

**-Q1.1-** Définir l'encapsulation.

**-Q1.2-** La déclaration suivante est-elle valide?

```
double f(double z=10.7, char* s);
```

vrai  faux

Dans le cas où c'est faux, pourquoi?

**-Q1.3-** Quel est le rôle d'un constructeur?

**-Q1.4-** Quand et pourquoi faut-il définir votre propre constructeur de copie, au lieu de se contenter du constructeur de copie par défaut?

**-Q1.5-** Quand un destructeur n'est **pas** appelé? (cochez une ou plusieurs réponses, pas besoin de justifications)

- a. une variable locale quitte sa portée
- b. delete
- c. delete [ ]
- d. aucune des précédentes réponses.

**-Q1.6-** Vers quoi pointe le pointeur "this"?

**-Q1.7-** Dans les déclarations suivantes, quels sont les expressions vraies, si Ptr1 et Ptr2 pointe vers le même objet, et pourquoi? (cochez une ou plusieurs réponses)

**a.** `Ptr1 == Ptr2`

vrai       faux

pourquoi?

**b.** `*Ptr1 == *Ptr2`

vrai       faux

pourquoi?

**c.** `*(Ptr1 == *Ptr2)`

vrai       faux

pourquoi?

**d.** `&Ptr1 == &Ptr2`

vrai       faux

pourquoi?

## Question 2 (10 points)

Que vont afficher les déclarations suivantes, en cas d'erreur de syntaxe, expliquez l'erreur et corrigez-la.

Représentez le retour à la ligne (retour chariot) par le signe ↵ :

a. `cout << "Message:\n\"Bonjour!\" \n";`

b. `cout << "Message:" << endl << "Bonjour!" << endl;`

c. `cout << "Message:\n\"Bonjour!\n";`

d. `cout << "Message:\n\"Bonjour!\"" << endl;`

e. `cout << "\Message:\" << endl << "\Bonjour!\" << endl;`

### Question 3 (20 points)

#### -Q3.1-

Tout en développant votre réponse sur la feuille d'examen, indiquez le résultat du programme suivant.

```
#include <iostream.h>

void f(int& x, int& y)
{
    x += 10;
    y -= 5;
}

int main()
{
    int x = 8;
    int y = 9;

    f(x,y);
    cout << x << ' ' << y << endl;

    f(y,x);
    cout << x << ' ' << y << endl;

    return 0;
}
```

**-Q3.2-**

Tout en développant votre réponse sur la feuille d'examen, indiquez le résultat du programme suivant.

```
#include <iostream.h>

void f(int& x, int* y) {
    x += 20;
    *y -= 10;
}

void f(int* x, int y) {
    *x += 20;
    y -= 10;
}

int main() {
    int x = 8;
    int y = 9;

    f(x,&y);
    cout << x << ' ' << y << endl;

    f(&y,x);
    cout << x << ' ' << y << endl;

    return 0;
}
```

#### Question 4 (20 points)

Soit le programme suivant:

```
#include <iostream.h>

class Z {
public:
    Z(int x) {
        v=x;
        cout << "constructeur" << endl;
    }
    Z(const Z& D) {
        v = D.v;
        cout << "recopie" << endl;
    }
    ~Z() {
        cout << "destructeur" << endl;
    }
private:
    int v;
};

int main() {

    Z r(3); // (a)
    Z *pz = new Z(18); // (b)

    delete pz; // (c)

    Z t(r); // (d)
    Z s = r; // (e)

    // r.v = 200; (f)

    return 0;
} // (g)
```

**-Q4.1-** Donner les lignes de la fonction main où sont appelés les constructeurs de Z, en indiquant à chaque fois le nom de l'objet créé (si plusieurs objets sont créés pour une même ligne, indiquez-les tous):

lignes	(a)	(b)	(c)	(d)	(e)	(f)	(g)
constructeurs							
Z::Z(int)							
Z::Z(const Z&)							

**-Q4.2-** Indiquer les lignes de la fonction main, où est appelé le destructeur de Z, en indiquant à chaque fois le nom de l'objet détruit (si plusieurs objets sont détruits pour une même ligne, indiquez-les tous).

(a)	(b)	(c)	(d)	(e)	(f)	(g)

**-Q4.3-** Indiquer ce qu'écrit le programme en sortie:

**-Q4.4-** Que se passe-t-il si on enlève les commentaires de la ligne **(f)** du programme précédent (détaillez votre réponse)?

## Question 5 (30 points)

La classe **Instrument** contient ce qui suit:

données membres private:

```
char* nom; // nom de l'instrument  
double prix; // le prix de l'instrument
```

fonctions membres public:

**affiche** : permet d'afficher les données membres de la classe **Instrument**.

**meme\_prix**: une fonction booléenne permettant de comparer si deux **Instruments** coûtent le même prix. Elle retourne **true**, si c'est vrai, sinon **false** dans le cas contraire.

**Q5.1** Écrire la définition de la classe **Instrument**, pour permettre les opérations décrites dans le **main** suivant:

```
int main() {  
  
    Instrument A("Saxophone", 750.80);  
  
    Instrument B = A;  
  
    B.affiche();  
  
    Instrument C("Guitare", 380.95);  
  
    if (C.meme_prix(B)){  
        cout << "la guitare/le saxophone, meme prix!\n";  
    } else {  
        cout << "la guitare/le saxophone, prix different!\n";  
    }  
  
    return 0;  
}
```

**Q5.2** Tout en tenant compte de la définition de votre fonction **affiche** de la classe **Instrument**, indiquez le résultat de l'affichage.







