IFT1166 - Session Automne, Intra

Mohamed Lokbani

IFT1166 - INTRA

Nom:___________________________________| Prénom(s):_____________________________|

Signature:_______________________________| Code perm:_____________________________|

Date:20 Octobre 1999

Durée: 2 heures (de 18h30: 20h:30) Local: 1360

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.
- L'examen compte **14** pages incluant celle-ci.

Nota Bene: 21 Octobre 1999: cette version est une remise à jour de la version distribuée aux étudiants durant l'intra. Cette version prend en compte les modifications apportées (très légères), durant l'Intra, à quelques questions de l'examen.

1.____________________/20

2.____________________/20

3.____________________/5

4.____________________/25

5.____________________/30

Total:__________________/100

Question 1 (20 points)

-1- Le message est le moyen unique de communiquer avec un objet. Que contient un tel message? (les citer uniquement)

Réponse :

-2- Définissez l’instanciation.

Réponse :

-3- Citez les 3 principes de base qui permettent de dire qu’un langage donné est orienté objet? (sans les expliquer)

Réponse :

-4- Quelles sont les deux façons d'écrire des commentaires en C++, et en quoi elles diffèrent l'une de l'autre?

Réponse :

-5- Quelle est la différence entre les membres privés et les membres publics, au niveau de la visibilité?
Réponse :

-6- Quel est le nom donné à la fonction membre appelée automatiquement lors de la création d'un objet?
Réponse :

-7- Quelle est la différence entre les 2 déclarations suivantes :

```
const int * A;
```

ET

```
int * const B;
```

Réponse :

Question 2 (20 points)

Le programme suivant utilise la macro MAX pour calculer le maximum entre deux nombres.

-1-
Indiquez le résultat de l'affichage, pour chaque fragments de code suivant:

```
#include <iostream.h>

#define MAX(a,b) ((a) > (b) ? (a) : (b))

int main () {
      int a =6,b=5,c=100;
      cout << "a: " << a << " b: " << b << " c: " << c << endl;
```

Réponse :

```
      c=MAX(a,b);
      cout << "a: " << a << " b: " << b << " c: " << c << endl;
```

Réponse :

```
      c=MAX(a++,b);
      cout << "a: " << a << " b: " << b << " c: " << c << endl;
```

Réponse :

```
      return 0;
}
```

-2-
Réécrivez la définition de la fonction MAX grâce a une fonction qui offre la rapidité de la macro ci-dessus, mais sans les possibilités de calculs incorrects.

Réponse :

-3-

À l'aide de cette nouvelle fonction, donnez le résultat d'affichage des deux fragments de code suivants:

```
c=MAX(a,b);
cout << "a: " << a << " b: " << b << " c: " << c << endl;
```

Réponse :

```
c=MAX(a++,b);
cout << "a: " << a << " b: " << b << " c: " << c << endl;
```

Réponse :

Question 3 (5 points)

Le fragment de code contient une erreur. Identifiez-la et expliquez la :

```
namespace exemple {
      void affiche(int ,int );
      // …
}

namespace test {
      void affiche(int );
      // ….
}

using exemple::affiche;

affiche(230);
```

Réponse :

Question 4 (25 points)

Le programme suivant gère un parc automobile.

-1-

Ce programme comporte 5 erreurs de compilation. Pour chaque erreur, indiquez sur quelle ligne elle apparaît et corrigez-la.

```
#include <iostream.h>
#include <string.h>

class voiture {
private:
        char *nom; // nom du vehicule
        int annee; // annee du vehicule
        static int total = 0; //nombre de vehicules
public:
        voiture(char* val="",int x=0); //constructeur initialisateur
        ~voiture(); // destructeur
        voiture(const voiture); // constructeur de recopie
        void affiche();
        static void print();
};



voiture::voiture(char* cp,int x)
{
        nom = new char[strlen(cp)+1];
        annee = x;
        strcpy(nom,cp);
        total++;
        cout << "C" << endl;
}

voiture::~voiture()
{
        delete nom;
        total--;
        cout << "D" << endl;
}

voiture::voiture(const voiture& car):nom(new char[strlen(car.nom)+1])
{
        strcpy(nom,car.nom);
        annee = car.annee;
        total++;
        cout << "R" << endl;
}

void voiture::affiche()
{
        cout << "nom : " << nom << "  annee: " << annee << endl;
        cout << "nombre de voitures dans le parc: " << total << endl;
}

void voiture::print()
{
        cout << "nombre de voitures dans le parc: " << total << endl;
        cout << "nom: " << nom << endl;
}

int main()
{
        voiture.print();
        voiture honda;
        voiture volkswagen("beatle",1985);
        voiture usagee("jetta",1980) = volkswagen;
        usagee.affiche();
        voiture::print();
        return 0;
}
```

Réponse :

Erreur 1:
Ligne:

Correction:

Erreur 2:
Ligne:

Correction:

Erreur 3:
Ligne:

Correction:

Erreur 4:
Ligne:

Correction:

Erreur 5:
Ligne:

Correction:

-2-

Supposez que le programme compile correctement, indiquez le résultat de l'affichage.

Réponse :

Question 5 (30 points)

Pour représenter un rationnel (à partir d'entiers), vous devez définir une classe contenant un numérateur entier et un dénominateur entier.

Un objet r de la classe Rationnel sera interprété comme représentant le nombre r_numerateur/r_denominateur. Par exemple, si l'on déclare:

```
Rationnel r1(2, 1), r2(6, 3), r3(4, 3);
```

alors les objets r1, r2 et r3 représentent respectivement 2/1, 6/3 et 4/3. r1 et r2 représentent alors le même nombre rationnel, mais r3 < r1 et r3 < r2.

Vous pouvez supposer que le dénominateur n'est pas nul et que le rationnel est tout le temps positif.

Nous désirons effectuer les opérations suivantes sur les deux rationnels r1(a,b) et r2(c,d):

- l'addition: r1+r2; retourne un rationnel.

L'addition de deux rationnels est définie ainsi:

r3 = r1 + r2 = ((a*d)+ (b*c)) / (b*d)

- la multiplication: r1*r2; retourne un rationnel.

La multiplication de deux rationnels est définie ainsi:

r3 = r1 * r2 = (a*c) / (b*d)

- le test d'égalité: r1==r2; retourne vrai, si r1=r2 (c'est à dire: a/b == c/d) ; sinon faux.
- le test de supériorité: r1>r2; retourne vrai si r1>r2 (c'est à dire: a/b > c/d), sinon faux.

En plus de ces opérations, il nous faudra une fonction pour afficher un rationnel sous la forme suivante:

r_numerateur/r_denominateur

Écrire la classe Rationnel et les fonctions membres citées plus haut, y compris le constructeur, pour pouvoir déclarer un rationnel sous la forme:

```
Rationnel r1(2, 1);
```

N'utilisez pas la technique de surcharge des opérateurs.

Réponse :