

4ETI IMI, Examen [C++]  
Programmation générique en C++  
CPE Lyon

2013-2014 (1ere session)  
durée 2h

Tous documents et calculatrices autorisés.

Répondez aux questions sur une copie séparée

Le sujet comporte 5 pages

*Le temps approximatif ainsi que le barème sont indiqués pour les grandes parties. Notez que le barème est donné à titre purement indicatif et pourra être adapté par la suite.*

*En cas de doute sur la compréhension de l'énoncé, explicitez ce que vous comprenez et poursuivez l'exercice dans cette logique.*

*Note préliminaire:*

- Toutes les questions concernent le langage de programmation C++ dans le cadre du développement de logiciels sur architecture PC standard.
- Nous vous invitons à commenter le code et les réponses. En particulier, en cas d'ambiguïté de compréhension d'une question, ajoutez toutes remarques et illustrations supplémentaires permettant d'expliquer votre démarche.
- Sauf mention contraire explicite, on supposera que l'on dispose d'un système Linux standard fonctionnant correctement sur un PC récent 32 ou 64 bits (identiques aux conditions de TP des PC de CPE: *int* étant encodé sur 4 octets, pointeurs étant encodés respectivement sur 4(/8) octets sur 32(/64) bits).
- On supposera que le code C++ est compilé avec une version récente de g++ sous la norme C++11 (ou ultérieure) identique aux conditions de TP des PC de CPE (avec l'option `-std=c++0x`).
- On supposera dans chaque cas que les `#include` des en-tête standards nécessaires à la bonne compilation et exécution des programmes décrits sont correctement installés et appelés (ex. `iostream`, `string`, etc).

## 1 Références

[15min max] 3 points

Soit le programme suivant:

```

1  int f1(int& x) { ++x; return x;}
2  int f2(const int& x) {return x+1;}
3
4  int main()
5  {
6    int a=5;
7    f1(a);
8    f1(2*a);
9    f1(5);
10   f2(a);
11   f2(2*a);
12   f2(5);
13   f2(f1(a));
14   f1(f2(a));
15 }
```

**Question 1** Indiquez (sur votre copie) pour chacune des lignes entre 7 et 14 si la ligne correspondante est valide ou bien génère une erreur de compilation.

## 2 STL

[25min max] 3 points

Soit le programme suivant.

```

1  struct index
2  {
3    int x,y;
4    index(int x_arg,int y_arg):x(x_arg),y(y_arg){}
5  };
6
7  struct comparator
8  {
9    bool operator()(const index& i0,const index& i1) const
10   {
11     return i0.x+N*i0.y < i1.x+N*i1.y;
12   }
13   const int N=5;
14 };
15
16 int main()
17 {
18   std::vector<std::map<index,float,comparator> > T(2);
19   T[0][index(4,1)]=1.4;
20   T[0][index(3,0)]=2.1;
21   T[0][index(3,2)]=-1.1;
22   T[1][index(0,0)]=4.2;
23
24   return 0;
25 }
```

**Question 2** *Ecrivez un parcours complet sur l'ensemble des valeurs contenus dans la variable T. On affichera ces valeurs dans l'ordre du vecteur puis de la map. On pourra utiliser les itérateurs. On supposera que ce parcours est écrit à partir de la ligne 23 de la fonction `main()`.*

**Question 3** *Écrivez la trace d'exécution correspondante, c'est à dire, l'ordre dans lequel les valeurs vont s'afficher.*

### 3 Classe et template

[40 min max] 7 points

On souhaite définir une classe pouvant désigner un nombre  $x$  irrationnel de manière exacte si il s'écrit sous la forme  $x = a + b\sqrt{n}$ , avec  $a$ ,  $b$  et  $n$  des nombres entiers.

On suppose pour cela que l'on définit une classe dont les attributs privés sont  $a$  et  $b$ .  $n$  est un paramètre template de la classe.

**Question 4** *Ecrivez la définition de cette classe que l'on appellera `nombre_exact`.*

On suppose que cette classe définit deux constructeurs. Un constructeur vide qui initialisera les valeurs  $a$  et  $b$  à 0, et un constructeur ou l'on passera deux paramètres entiers initialisant  $a$  et  $b$ .

**Question 5** *Ecrivez les deux constructeurs décrits.*

**Question 6** *Donnez la déclaration des variables  $x_0 = 1 + \sqrt{2}$  et  $x_1 = 1 + \sqrt{3}$  tels que vous pourriez les déclarer dans la fonction `main()` par exemple.*

Soit  $x_0$  un nombre de type `nombre_exact` valant  $2 + 3\sqrt{2}$ .

**Question 7** *On souhaite que l'appel à `std::cout<<x0<<std::endl`; affiche en ligne de commande  $2 + 3 * \text{sqrt}(2)$ . Ecrivez la fonction permettant de réaliser cela.*

**Question 8** *On souhaite que l'appel à `x0()` renvoie la valeur approchée de ce nombre sous la forme d'un flottant. Donnez l'en-tête de la fonction ou méthode permettant de réaliser cela. Sous quelle forme est-il le plus judicieux de définir le paramètre de retour: `float`, `float&`, ou `const float&`?*

Supposons que  $x_1$  soit un nombre de type `nombre_exact` valant  $3 + 6\sqrt{2}$ .

**Question 9** *On souhaite que les opérations suivantes soient permises:*

$x_0 += x_1; x_0 + x_1; 4 * x_0;$

*Donnez les en-têtes des fonctions ou méthodes permettant de réaliser ces opérations.*

**Question 10** *On souhaite placer les nombres  $x_0$ ,  $x_1$ , et  $x_2 = 7 - 3\sqrt{3}$  dans un conteneur de type `std::vector`. Est-ce possible? Si oui, écrivez le code correspondant. Si non, expliquez pourquoi, et décrivez quelle approche permettrait de réaliser cela.*

**Question 11** *Lorsque le paramètre template  $n$  vaut 0, il est inutile de stocker la valeur correspondant à la partie irrationnelle. Afin de pouvoir économiser en espace mémoire, définissez une spécialisation de la classe `nombre_exact` qui ne stocke pas ce paramètre lorsque  $n$  est nul.*

## 4 Héritage

[40 min max] 7 points

On donne une classe `Menu` spécialisée dans la gestion d'un menu en mode texte. Elle prend en charge l'affichage du menu, la saisie du choix de l'utilisateur, et l'exécution du traitement correspondant au choix de l'utilisateur. Cette classe utilise des objets `OptionMenu` munis d'une méthode `Executer` qui est appelée par l'objet `Menu` lors du choix de cette option. Pour utiliser cette classe, il faut donc définir des classes dérivées d'`OptionMenu` et les ajouter dans le menu. La classe `OptionQuitter` est définie pour gérer la sortie de l'application. Voici la déclaration et la définition de ces différentes classes :

```

class OptionMenu
{
    std::string Nom;

public:

    OptionMenu(const std::string& ch) : Nom(ch) {}
    virtual ~OptionMenu() {}

    const std::string &LireNom() const {return Nom;}
    virtual bool Executer() const=0;
};

class Menu : public std::vector<OptionMenu*>
{
public:
    void Afficher() const;
    void Executer() const;
};

void Menu::Afficher() const
{
    std::cout<<"Menu: ";<<std::endl;
    for(unsigned int k=0;k<size();++k)
        std::cout<<"- "<<k<<" : "<<at(k)->LireNom()<<std::endl;
}

void Menu::Executer() const
{
    bool fin=false;
    while(!fin)
    {
        Afficher();
        std::cout<<"Choix ? "<<std::endl;
        unsigned int choix=-1;

        std::cin>>choix;
        if(choix<size())
            fin=at(choix)->Executer();
        else
            std::cout<<"Choix invalide"<<std::endl;
    }
}

```

```

class OptionQuitter : public OptionMenu
{
public:
    OptionQuitter() : OptionMenu("Quitter") {}

    bool Executer() const;
};

bool OptionQuitter::Executer() const
{
    std::cout<<"Voulez-vous vraiment quitter (o/n) ?"<<std::endl;

    char reponse;
    std::cin>>reponse;
    if(reponse=='o')
        return true;
    return false;
}

```

**Question 12** Schématisez graphiquement la relation entre les différentes classes de l'application (diagramme des classes). Expliquez pourquoi la méthode `Executer` de la classe `OptionMenu` est virtuelle pure.

**Question 13** Ecrire une fonction principale (`main`) permettant d'utiliser la classe `Menu` avec la seule option `Quitter`. Donnez un résultat d'exécution.

On souhaite désormais réaliser une classe `OptionEssai` permettant de gérer une option notée *Essai* dans le menu proposé. Lorsque l'on choisira *Essai* dans le nouveau menu, on affichera simplement la phrase *Vous avez selectionne l'option essai*.

**Question 14** Déclarez et définir en C++ la classe `OptionEssai` ainsi que ses méthodes. Donnez la nouvelle fonction principale permettant d'utiliser cette classe.