

Examen C++ - ISIMA 2 - 06/12/2011

2h – Sans document sauf feuille manuscrite R/V autorisée

Téléphone interdit

Soigner la présentation, des points sont accordés à la présentation et à la lisibilité

Le barème est donné à titre indicatif

Le sujet comporte 4 pages

Questions (7 pts)

Question 1 (1,5 pt)

Donner des exemples de code C++ pour les questions suivantes et préciser le cycle de vie (création, destruction des objets). On distinguera les 3 cas suivants :

- 1) Allocation statique d'un objet (considéré comme une variable globale).
- 2) Allocation automatique d'un objet (variable locale d'une méthode par exemple).
- 3) Allocation dynamique d'un objet.

Question 2 (1,5 pt)

Soit le code :

```
int *ptr = new int (10) ;  
for (int i = 0 ; i < 10 ; i++)  
    cin >> ptr[i] ;
```

- 1) Que souhaite réaliser le programmeur ?
- 2) Est-ce que ce code compile sans problème ? Que va réaliser réellement le code proposé ?
- 3) Quelle modification proposez-vous pour un fonctionnement plus cohérent ?

Question 3 (2 pts)

En C++, le mot clé "const" peut être utilisé de différentes façons.

- 1) Donner les différentes utilisations de ce mot clé.
- 2) Pourquoi est-il conseillé de l'utiliser autant que possible ?

Question 4 (2 pts)

```
#include <iostream>  
  
class VehiculeMotorise  
{  
    private :  
        int    puissanceMoteur_ ;  
  
    public :  
        VehiculeMotorise( int inPuissanceMoteur )  
            : puissanceMoteur_( inPuissanceMoteur )  
        {}  
  
        virtual void afficher() const  
        {  
            std::cout << "Puissance moteur de : " << puissanceMoteur_ << std::endl ;  
        }  
};
```

Étudiez le code précédent. Pour chaque appel de fonction (de a à l), indiquez si le code compile ou pas.

- S'il compile, donnez la sortie correspondante (1 ou 2) et expliquez brièvement le comportement menant à cette sortie.
- Sinon indiquez quelles sont les raisons de l'erreur de compilation.

Problème (13 pts)

Consignes générales :

- Rédiger sur une **copie séparée**.
- Dès que cela est justifié, les méthodes implémentés devront être définis constantes.
- Lorsque l'implémentation d'une classe est demandée, seules les méthodes nécessaires devront être implémentées.

Le père Noël nous demande d'aider à organiser le système d'information de fabrication et de distribution des jouets de Noël.

Tous les jouets à distribuer sont stockés au pôle nord. Ils sont tout d'abord stockés dans un premier entrepôt par type dans d'énormes étagères. Les lutins préparent ensuite des sacs pour la distribution où chaque sac est destiné à une zone géographique bien précise.

1- Tous les jouets sont repérés par la technologie RFID : ils ont tous un identifiant RFID **donné** (on ne vous demande pas de le calculer ni même de s'assurer de son unicité) lors de leur arrivée au pôle nord. On connaît également l'âge minimum de l'enfant qui doit l'utiliser, le volume du jouet, l'année de fabrication. Le nom de l'enfant qui va le recevoir est inconnu au départ et fixé par les lutins au moment de la préparation du sac.

Écrire la déclaration de la classe Jouet avec toutes les méthodes nécessaires inline. Le constructeur utilisera une liste d'initialisation pour tous les paramètres. (1,5pt).

2- Les lutins aiment avoir des statistiques sur leur travail : tous les jouets qui arrivent au pôle nord sont comptés, ce compte est à tout moment disponible par chacun et le père Noël (et seulement lui) doit avoir la possibilité de changer ce nombre quand il le désire (en début de campagne par exemple ...).

Ajouter ce qu'il faut à la classe Jouet pour avoir ces comportements. Expliquer vos choix, donner ce qui doit être dans un fichier d'entête ou au contraire dans un fichier d'implémentation. (2 pts)

3- Voici quelques types de jeu et des exemples de jeux pour chacun de ces types : jouets de plein air (ex : vélo, trottinette, toboggan), jouets de construction (ex : cubes en bois, pièces magnétiques, trains), jouets d'imagination (ex : playmobil, barbie, ferme), de réflexion (ex : casse-tête, bataille navale, trivial pursuit).

Les jeux de plein air indiquent clairement une durée de vie, les jeux de construction le nombre de pièces, les jouets d'imagination une catégorie spécifiée dans une énumération : ville, ferme, antiquité, SF, guerre, les jeux de réflexion un nombre minimum et maximum de joueurs.

- a) Nous devons avoir l'information de la catégorie de jouet (une chaîne de caractères par type) sur un jouet même si cela n'a pas de sens pour une instance de Jouet. Quel impact, cela a-t-il sur la classe Jouet, comment est-ce codé ? (1,5 pt)
- b) Écrire les classes nécessaires à l'instanciation d'un objet de la classe Vélo, chaque vélo ayant une couleur. Écrire également l'instanciation d'un tel Vélo. (1,5pt)

```

class Voiture : public VehiculeMotorise
{
private :
    int     nombrePortes_;

public :
    Voiture( int inPuissanceMoteur, int inNombrePortes )
        : VehiculeMotorise( inPuissanceMoteur ),
          nombrePortes_( inNombrePortes )
    {}

    virtual void afficher() const
    {
        VehiculeMotorise::afficher();
        std::cout << "Nombre de portes : " << nombrePortes_ << std::endl;
    }
};

void afficherVehiculeParValeur( VehiculeMotorise inVehiculeMotorise )
{ inVehiculeMotorise.afficher(); }
void afficherVehiculeParRef( VehiculeMotorise & inRefVehiculeMotorise )
{ inRefVehiculeMotorise.afficher(); }
void afficherVehiculeParPtr( VehiculeMotorise * inPtrVehiculeMotorise )
{ inPtrVehiculeMotorise->afficher(); }

void afficherVoitureParValeur( Voiture inVoiture )
{ inVoiture.afficher(); }
void afficherVoitureParRef( Voiture & inRefVoiture )
{ inRefVoiture.afficher(); }
void afficherVoitureParPtr( Voiture * inPtrVoiture )
{ inPtrVoiture->afficher(); }

int main( int, char *[] )
{
    // Manipulation de Voiture
    Voiture     maPremiereVoiture( 100, 5 );

    /* a */ afficherVehiculeParValeur( maPremiereVoiture );
    /* b */ afficherVehiculeParRef( maPremiereVoiture );
    /* c */ afficherVehiculeParPtr( &maPremiereVoiture );

    /* d */ afficherVoitureParValeur( maPremiereVoiture );
    /* e */ afficherVoitureParRef( maPremiereVoiture );
    /* f */ afficherVoitureParPtr( &maPremiereVoiture );

    // Manipulation de VehiculeMotorise
    VehiculeMotorise     monPremierVehicule( 100 );

    /* g */ afficherVehiculeParValeur( monPremierVehicule );
    /* h */ afficherVehiculeParRef( monPremierVehicule );
    /* i */ afficherVehiculeParPtr( &monPremierVehicule );

    /* j */ afficherVoitureParValeur( monPremierVehicule );
    /* k */ afficherVoitureParRef( monPremierVehicule );
    /* l */ afficherVoitureParPtr( &monPremierVehicule );

    return 0;
}

```

Sorties possibles :

(1)
 Puissance moteur de : 100
 Nombre de portes : 5

(2)
 Puissance moteur de : 100

- c) Faites de même pour une boîte de jeu Lego qui est un jeu à la fois de construction et d'imagination. (1,5 pt)
- 4- Les sacs que manipulent les lutins peuvent contenir des jouets. Les lutins les remplissent jusqu'à ce qu'ils soient pleins. Les sacs ont un volume fixé à l'instanciation.
- Écrire la classe Sac comprenant entre autre une méthode permettant de connaître le volume utilisé par les objets du sac et une méthode permettant de savoir si le sac est plein. On utilisera pour cela un conteneur de la STL. (1,5 pt)
 - Écrire l'opérateur de redirection de flux << qui utilisera la fonction `std::copy` pour afficher le contenu d'un sac. Précisez ce qu'il faut éventuellement ajouter aux autres classes pour que cela fonctionne ? (1pt)
- 5- On veut écrire une méthode `ranger()` dans la classe Lutin qui prend en paramètre le nom d'un enfant, un Jouet qui lui est destiné, et un Sac où stocker le jouet. Écrire une telle fonction qui fixe le destinataire du jouet, met le jouet dans le sac et affiche sur la sortie standard le type correct du jouet (on suppose que la classe Sac dispose, en plus des méthodes déjà définies, d'une méthode `ajouterJouet` qui ajoute au sac un jouet passé en paramètre). (1pt)
- 6- La liste de distribution du père Noël est une liste qui associe une zone géographique de distribution (stockée sous forme d'entier) avec la liste des sacs à distribuer.
- Écrire l'instanciation de la liste du père Noël en utilisant la STL. (0,5 pt)
 - Afficher la liste de tous les enfants qui vont recevoir un cadeau. (1 pt)