QUESTION I.E 2

Qu'invoque le constructeur par default d'une classe composé ?

Le constructeur par défaut des classes composantes.

FEUILLE 4

Q1 / Pour quelle raison le constructeur de copie n'est il pas déclaré dans la classe ?

Parce que le compilateur va le générer automatiquement. Attention si on le rajoute dans la classe, le compilateur va penser qu'on veut le surcharger et va chercher la surcharge.

Q2 / Pour quelle raison le destructeur n'est il pas déclaré dans la classe ?

Pour la même raison.

Q3 / Pour quelle raison l'Opérateur de Résolution de Portée figure en tête des définitions de méthodes ?

Pour préciser au compilateur dans quelle classe se trouve la méthode qui suit.

Q4 / Une méthode *inline* peut elle faire l'objet d'une compilation séparée ?

Non, une méthode inline ne peut pas être séparée de sa définition.

Q5 / Que désigne la méta notation *this* ?

C'est un pointeur implicite qui pointe sur l'objet support, moi-même.

FEUILLE 6

Q1 / v3= v1;

<-> v3.operator = (v1) ; en C : operator = (v3, v1) ;

Q2 / v3= -v2;

<-> v3.operator = (v2.operator – ());

Q3 / return v1 == v2;

<-> return v1.operator == (v2); retourne un booléen true si les deux valeurs sont égales sinon false.

Q4 / cout << v3;

<-> cout.operator << (cout, v3);

Q5 / v3= v1 + v2;

<-> v3.operator = (v1.operator + (v2)) ;

TP 9

Q1 / Pour quelle raison interdit on la duplication de toute collection ?

Pour éviter la double vue car on ne sait pas dupliquer l'objet pointé car void *. Problème au niveau de la destruction. Pas d'opérateur + car sinon on doit dupliquer la collection de base, donc double vue.

Q2 / Pour quelle raison la signature formelle de la méthode *toString* spécifie t-elle un paramètre ?

La méthode toString ne sait pas à quoi correspond le void * donc elle ne peut connaître la valeur.

Paramètre pour qu'on puisse lui passer une fonction pour connaître la valeur à afficher.

Le module de test unitaire est l'utilisateur de la fonction donc c'est elle qui fournit la méthode toString qui connaît le type.

Q3 / Quel est la nature de ce paramètre ?

Il s'agit d'un pointeur de fonction.

Q4 / Quel est l'usage de ce paramètre ?

Il permet d'aller chercher la bonne méthode toString au bon moment.

Le pointeur de fonction sert à transtyper ce qu'on a dans la collection.

Q5 / Quel est le rôle du service *vider* ?

Son rôle est de détruire le pointeur sur l'objet pointé.

Vider supprime tous les éléments de la collection mais conserve le conteneur.

C'est le destructeur qui se charge de supprimer le conteneur.

Le destructeur est appelé à l'accolade fermante du bloc qui déclare la variable correspondante.

Q6 / Quelle est la définition du mécanisme de "*transtypage dynamique implicite ascendant*" ?

Analyser soigneusement les codes sources du constructeur de copie.

Quelles sont les lignes qui exploitent le mécanisme précité ?

Cardinal()…

Q7 / Quelles restrictions d'usage imposent ce mécanisme pour l'objet transtypé ?

L'objet transtypé ne peut plus utiliser ses attributs et méthodes propres, il est devenu une mère.

La classe *Ensemble* - Version 1.0.0 / Surcharges des opérateurs += et -=

Justifier la surcharge de ces deux opérateurs.

On veut contrôler qu'il n'y ai aucun élément commun dans l'ensemble. Vérifier l'appartenance.

TP 10

Exercice 3)

Q1 / Dans la nouvelle gestion de l'historique des opérations bancaires, pour quelle raison technique ne peut on pas utiliser une instance de la classe *Ensemble* ?

Car ce n'est pas un ensemble (l'unicité n'est pas respectée).

Exercice 4)

Q1 / Que signifie l'expression "transtypage dynamique explicite"

Pouvoir changer de type au moment de la compilation.

Q2 / Dans quel cas ce transtypage est il valide ?

En C il faut que les types soient compatibles. En C++ tout objet peut se transtyper dans la classe de ses ascendants.

Valide dans deux cas :

- Liens estain
- Lorsque le cast est possible (void*) notion de généricité.

Q3 / Identifier un exemple de mise en oeuvre du mécanisme, rencontré dans l'exercice ci-dessus.

Dans dateDerniereOperation

Q4 / Quelles vérifications effectue le compilateur lors de la mise en oeuvre du mécanisme ?

S'il y a bien une notion de hiérarchisation d'héritage ou type générique.

Aucune, si ce sont des objets et des classes.

Q5 / Peut on transtyper un pointeur sur "compte courant" en un pointeur sur "compte épargne" ?

Oui, bien qu'on ne satisfasse pas les règles précédentes.

TP 11

Exercice 7)

Q1 / Une surcharge de l'opérateur += aurait elle été possible en lieu et place de la méthode *ouvrirCompte* ?

La méthode += ne prend qu'un paramètre donc sa aurai marché pour le compte courant mais pas pour le compte épargne (deux paramètres).

Question de cours

Q1 / Un constructeur d'une classe C++ peut il invoquer un autre constructeur de la même classe ?

Non

Q2 / Justifier le choix d'une classe abstraite.

Il n'y a qu'un seul et unique simulateur pas besoin d'instancier.

Q5 / Quel est le rôle de la méthode c_str() de la classe string ?

Elle rend le "char*" qui est dans le string.

Q2 / Justifier l'intérêt technique de l'instruction C enum et ses limites ?

Cela permet de définir toutes les possibilités d'une variable.

"friend" permet de prendre attribut et méthodes privé et publique de la classe concernée.

RTTI

Q1 / Que signifie l'abréviation anglaise R.T.T.I. ?

Real Time Type Identification.

Q2 / Quelle est la traduction en français ?

Identification dynamique de type.

Q3 / Détailler le fonctionnement du mécanisme.

Garder en mémoire le type d'un objet à sa naissance.

Q4 / Quel est le rôle de l'instruction C++ *type_id* ?

Elle va nous permettre d'accéder au RTTI de l'objet ou du type prédéfinit. Equivalent à dynamic_cast.

Q5 / Que représente le mot clé *type_info* du langage C++ ?

C'est le type prédéfinit de retour de type_id, on exécute dessus l'instruction name pour connaître le type.

RTTI en une phrase :

Connaitre le type de l'objet a l'exécution

Détailler soigneusement la mise en œuvre de ce mécanisme pour un programmeur C++ :

Un pointeur caché qui va pointer sur la carte d'identité

Pour quelle raison fonctionnelle, les surcharges des opérateurs [] et ( ) doivent elles être privées ?

Car on ne doit pas accéder aux données personnelles propres à un client.