**Contrôle Continu AP4A – Vendredi 7 Mai 2021 – durée 1h**

**I.** (7pts) Indiquer quelles sont les erreurs de compilation lorsqu’il en existe dans les exemples C++ suivants. Dans tous les cas de figure, expliciter la réponse dans la case commentaire.

**1)**

class B {

protected:

 int i;

 friend void f();

};

class D : public B {};

void f() {

 B\* p = new B;

 D\* q = new D;

 int fi1 = p->i;

 int fi2 = q->i;

}

**Commentaire**

**2)**

class B {

protected:

 int i;

};

class D : public B {};

void f() {

 B\* p = new B;

 D\* q = new D;

 int fi1 = p->i;

 int fi2 = q->i;

}

**Commentaire**

**3)**

class A {

protected:

 int a;

};

class B : public A {};

class C : public B {

 void f(B\* p);

};

void C::f(B\* p) {

 a = 1;

 p->a = 2;

}

**Commentaire**

**4)**

class A {

protected:

 int a;

};

class B : public A {void f();};

class C : private B {void f();};

void B::f() {a = 1;}

void C::f() {a = 1;}

**Commentaire**

**5)**

class A {

protected:

 int a;

};

class B : private A {void f( );};

class C : public B {void f( );};

void B::f() {a = 1;}

void C::f() {a = 1;}

**Commentaire**

**6)**

class B {

protected:

 int a;

};

class D : public B {

 friend void f( );

public:

 int b;

};

void f() {

 D\* p;

 p->a = 1;

 p->b = 2;

 B\* pp;

 pp->a = 1;

 pp->b = 1;

}

**Commentaire**

**7)**

class A {

protected:

 int a;

};

class B : protected A {

 void f();

};

class C : private B {

 void f();

};

void B::f() {a = 1;}

void C::f() {a = 1;}

**Commentaire**

**II.** (3pts) Considérons le type Vecteur en C++ tel que réalisé en TP et des variables de type Vecteur. Donner la notation fonctionnelle équivalente des instructions suivantes :

a = a;

c = a + b;

v[i] = v[j] = 2;

a = b = c = 2;

cout << u << “ “ << v << endl;

((a = b) = c) = 2;

**III. (10 pts)** Questions diverses

*Question 1*

Soient une classe A et a1 une instance de cette classe. Quelle est la signification de l’instruction suivante ? (préciser les méthodes et/ou constructeurs qui sont invoqués)

A a2 = a1 ;

*Question 2*

Soient deux classes distinctes A et B. Dans quel cas la ligne de code suivante a-t-elle un sens ?

A a ; B b ; b=a ;

*Question 3*

Quelles sont les différences qui existent entre héritage public et héritage privé conceptuellement ?

*Question 4*

On définit les classes suivantes :

#include <iostream.h>

class ObjetGraphique {

 protected : int x,y ;

 ObjetGraphique \*suiv ;

 public : void dessine(){}; };

class Cercle : public ObjetGraphique {

 private: int r ; // rayon

 public: void dessine() { cout << "Cercle" << endl ; } };

class Rectangle : public ObjetGraphique {

 private: int l,h ; // longueur, hauteur

 public: void dessine() { cout << "Rectangle" << endl ; } };

class ObjetCompose : public ObjetGraphique {

 private: ObjetGraphique \*debut, \*fin ; // Liste d'objets graphiques

 public: ObjetCompose():debut(NULL), fin(NULL) {}

 void ajoute(ObjetGraphique \*o) {

 if (debut==NULL) { debut = fin = o ; }

 else { fin->suiv = o ; fin = fin-> suiv ; }

 fin->suiv = NULL ;

 }

 void dessine(){

 cout << "Objet Compose " << endl ;

 ObjetGraphique \*cour = debut ;

 while (cour!=NULL) { cour->dessine() ; cour = cour->suiv ; } } } ;

Quel affichage donne l’exécution du code suivant ? pourquoi ? quel est le problème ?

main (){

 ObjetGraphique \*c = new Cercle() ;

 ObjetGraphique \*r = new Rectangle() ;

 ObjetCompose \*oc = new ObjetCompose() ;

 (\*oc).ajoute(c) ;

 (\*oc).ajoute(r) ;

 ObjetCompose \*oc2 = new ObjetCompose() ;

 (\*oc2).ajoute(c) ;

 (\*oc2).ajoute(r) ;

 (\*oc).ajoute(oc2) ;

 oc->dessine() ;

 return (0) ;

}

*Question 5*

Quelle solution peut-on envisager pour remédier au problème évoqué dans la question précédente ?

*Question 6*

Faut-il définir la forme de Coplien pour la classe ObjetCompose étant donné qu’elle contient des pointeurs ?

*Question 7*

Que se passe-t-il lorsqu'on retourne dans une fonction une variable locale et que la fonction a pour type de retour une référence ? Que faut-il faire pour solutionner ce problème ?

object& invalid\_function()

{

 object the\_object\_to\_return;

 // ...

 return the\_object\_to\_return;

}

*Question 8*

Soit la classe d'entier Integer suivante :

 #include<iostream.h>

 class Integer {

 private:

 int i;

 public:

 Integer() { i=0;}

 Integer(int j) { i=j;}

 Integer(Integer& j) { i=j.i;}

 Integer& operator=(Integer& j) { i=j.i; return \*this ;}

 void affiche() { cout << i ; }

 } ;

1. Est-ce que l’écriture de la classe Integer est conforme aux conventions de codage C++ ? détailler et remédier.
2. L’écriture de l’opérateur d’affectation a-t-elle une utilité ici ? ou encore celle du constructeur par copie ?
3. Quel est le problème avec le code suivant ? détailler, compléter la classe Integer comme il faut.

 int main() {

 Integer i(2), j(3) ;

 Integer k = i+j ;

 k.affiche() ;

 return 0 ;

 }

1. Donner le profil et corps de l’opérateur de sortie « << » pour la classe Integer.