

[Tableau de bord](#) / [Mes cours](#) / [INF1010 - Programmation orientée objet](#) / [Quizzes Comptabilisés](#) / [Quiz 6-Hiver2023](#)

Commencé le vendredi 24 février 2023, 22:47

État Terminé

Terminé le vendredi 24 février 2023, 22:58

Temps mis 11 min 9 s

Points 14,00/14,00

Note 10,00 sur 10,00 (100%)

Question 1

Correct

Note de 1,00
sur 1,00

Une classe interface est une classe abstraite qui ne contient que des méthodes virtuelles pures qui devront être implémentées par la classe qui en dérive si on veut pouvoir l'instancier.

Veuillez choisir une réponse.

☒ Vrai ✓

☐ Faux

La réponse correcte est « Vrai ».

Question 2

Correct

Note de 2,00
sur 2,00

Soit le code suivant:

```
1 class A {
2     public:
3         A() {}
4         ~A() {}
5         void f1() {}
6         void f3() {}
7         void f4() {}
8         void f5() {}
9 };
10 class B : public A {
11     public:
12         B() {}
13         ~B() {}
14         void f3() {}
15         void f2() {}
16         void f4() {}
17         void f5() {}
18 };
19 class C : public A {
20     public:
21         C() {}
22         ~C() {}
23         void f2() {}
24         void f3() {}
25         void f4() {}
26 };
27 class D : public B, public C {
28     public:
29         D() {}
30         ~D() {}
31         void f2() {}
32         void f3() {}
33         void f4() {}
34         void f5() {}
35 };
```

28	};		
29	int main() {		
30			
31	D d;	Aucune erreur de compilation	✓
32			
33	d.f1();	Problème du diamant	✓
34			
35	d.f2();	Ambiguïté de nom	✓
36			
37	d.f3();	Aucune erreur de compilation	✓
38			
39	d.f4();	Ambiguïté de nom	✓
40			
	d.f5();	Problème du diamant	✓
	}		

Donnez le problème lié à l'héritage multiple sous-jacent à l'erreur de compilation, s'il y a lieu, pour chaque ligne du main.

Votre réponse est correcte.

La réponse correcte est : Soit le code suivant:

```

1 class A {
2 public:
3     A() {}
4     ~A() {}
5     void f1() {}
6     void f3() {}
7     void f4() {}
8     void f5() {}
9 };
10 class B : public A {
11 public:
12     B() {}
13     ~B() {}
14     void f3() {}
15     void f2() {}
16     void f4() {}
17     void f5() {}
18 };
19 class C : public A {
20 public:
21     C() {}
22     ~C() {}
23     void f2() {}
24     void f3() {}
25     void f4() {}
26 };
27 class D : public B, public C {
28 public:
29     D() {}
30     ~D() {}
31     void f3() {}
32 };
33 int main() {
34     D d; [Aucune erreur de compilation]
35     d.f1(); [Problème du diamant]
36     d.f2(); [Ambiguïté de nom]
37     d.f3(); [Aucune erreur de compilation]
38     d.f4(); [Ambiguïté de nom]
39     d.f5(); [Problème du diamant]
40 }
```

Donnez le problème lié à l'héritage multiple sous-jacent à l'erreur de compilation, s'il y a lieu, pour chaque ligne du main.

Question 3

Correct

Note de 6,00
sur 6,00

Soit la définition de la classe GestionnaireElements suivante:

```
template<typename T>
class GestionnaireElements {
public:
    GestionnaireElements(const vector<T>& init);
    template<typename Comp>
    void trierElements();
    void afficher() const;

private:
    vector<T> elements_;
};
```

Compléter l'implémentation de la méthode `trierElements()`:

```
template<typename T>
template<typename Comp>
void GestionnaireElements<T>::trierElements() {
    // Trie les éléments en fonction du comparateur Comp
}
```

Répondez aux questions suivantes:

1. Vrai ou Faux. Il faudra absolument expliciter le type générique T de la classe `GestionnaireElements` lorsqu'on instancie un objet de cette classe. ✓
2. Vrai ou Faux. Il faudra absolument expliciter le type générique Comp lorsqu'on instancie un objet de la classe `GestionnaireElements`. ✓
3. Vrai ou Faux. Il faudra absolument expliciter le type générique T de la classe `GestionnaireElements` lorsqu'on fait appel à la méthode `trierElements()` sur un objet de la classe `GestionnaireElements`. ✓
4. Vrai ou Faux. Il faudra absolument expliciter le type générique Comp lorsqu'on fait appel à la méthode `trierElements()` sur un objet de la classe `GestionnaireElements`. ✓

Votre réponse est correcte.

La réponse correcte est :

Soit la définition de la classe `GestionnaireElements` suivante:

```
template<typename T>
class GestionnaireElements {
public:
    GestionnaireElements(const vector<T>& init);
    template<typename Comp>
    void trierElements();
    void afficher() const;

private:
    vector<T> elements_;
};
```

Compléter l'implémentation de la méthode `trierElements()`:

```
[template<typename T>]
[template<typename Comp>]
void [GestionnaireElements<T>]::trierElements() {
```

```
// Trie les éléments en fonction du comparateur Comp
}
```

Répondez aux questions suivantes:

1. Vrai ou Faux. Il faudra absolument expliciter le type générique T de la classe **GestionnaireElements** lorsqu'on instancie un objet de cette classe. [Vrai]
2. Vrai ou Faux. Il faudra absolument expliciter le type générique Comp lorsqu'on instancie un objet de la classe **GestionnaireElements**. [Faux]
3. Vrai ou Faux. Il faudra absolument expliciter le type générique T de la classe **GestionnaireElements** lorsqu'on fait appel à la méthode trierElements() sur un objet de la classe **GestionnaireElements**. [Faux]
4. Vrai ou Faux. Il faudra absolument expliciter le type générique Comp lorsqu'on fait appel à la méthode trierElements() sur un objet de la classe **GestionnaireElements**. [Vrai]

Question 4

Correct

Note de 3,00
sur 3,00

Modifiez le code déjà écrit afin de rendre la classe Point générique (ses attributs seront d'un type T). Ne changez pas les noms des attributs et laissez-les public.

Nous voulons qu'un code comme le suivant ait deux points dont les types des attributs sont différents (on compile en C++17):

```
Point pointInt(3, 4);
Point pointDouble(3.5, 4.5);
```

Réponse : (régime de pénalités : 0 %)

Réinitialiser la réponse

```
1  template<typename T>
2  class Point
3  {
4  public:
5      Point(T i, T j): i_(i), j_(j) {}
6
7      T i_;
8      T j_;
9  };
10
```

	Test	Résultat attendu	Résultat obtenu	
✓	Point p(3, 4); cout << is_same_v<decltype(p.i_), int> << is_same_v<decltype(p.j_), int>;	11	11	✓
✓	Point p(3, 4); cout << (p.i_ == 3) << (p.j_ == 4);	11	11	✓
✓	Point p(3.5, 4.5); cout << is_same_v<decltype(p.i_), double> << is_same_v<decltype(p.j_), double>;	11	11	✓

✓	Point p(3.5, 4.5); cout << (p.i_ == 3.5) << (p.j_ == 4.5);	11	11	✓
---	---	----	----	---

Tous les tests ont été réussis ! ✓

Solution de l'auteur de la question (Cpp):

```

1  template <typename T>
2  class Point
3  {
4  public:
5      Point(T i, T j): i_(i), j_(j) {}
6
7      T i_;
8      T j_;
9  };
10 
```

Correct

Note pour cet envoi : 3,00/3,00.

Question 5

Correct

Note de 2,00
sur 2,00

Soit les classes Chercheur, Employe, Professeur et ProfInfo suivantes:

```

class Chercheur {
public:
    virtual void faireRecherche() = 0;
    virtual void publier() = 0;
};

class Employe {
public:
    Employe(string matricule, string nom);
    virtual void travailler() = 0;
private:
    string matricule_;
    string nom_;
};

class Professeur: public Chercheur, public Employe {
public:
    Professeur(string matricule, string nom);
    void travailler() override;
    void publier() override;
};

class ProfInfo: public Professeur {
public:
    ProfInfo(string matricule, string nom);
    void faireRecherche() override;
    virtual ~ProfInfo() = default;
};

```

Dites quel concept représente chacune des classes:

1. ProfInfo - Classe concrète ✓
2. Professeur - Classe abstraite ✓
3. Employe - Classe abstraite ✓
4. Chercheur - Interface ✓

Votre réponse est correcte.

La réponse correcte est :

Soit les classes Chercheur, Employe, Professeur et ProfInfo suivantes:

```
class Chercheur {  
public:  
    virtual void faireRecherche() = 0;  
    virtual void publier() = 0;  
};  
  
class Employe {  
public:  
    Employe(string matricule, string nom);  
    virtual void travailler() = 0;  
private:  
    string matricule_;  
    string nom_;  
};  
  
class Professeur: public Chercheur, public Employe {  
public:  
    Professeur(string matricule, string nom);  
    void travailler() override;  
    void publier() override;  
};  
  
class ProfInfo: public Professeur {  
public:  
    ProfInfo(string matricule, string nom);  
    void faireRecherche() override;  
    virtual ~ProfInfo() = default;  
};
```

Dites quel concept représente chacune des classes:

1. ProfInfo - [Classe concrète]
2. Professeur - [Classe abstraite]
3. Employee - [Classe abstraite]
4. Chercheur - [Interface]

◀ Quiz 5 -Hiver2023

Aller à...

Quiz 7-Hiver2023 ▶