

[Tableau de bord](#) / [Mes cours](#) / [INF1010 - Programmation orientée objet](#) / [Quizzes Comptabilisés](#) / [Quiz 3-Hiver2023](#)

**Commencé le** vendredi 3 février 2023, 20:40

**État** Terminé

**Terminé le** vendredi 3 février 2023, 20:42

**Temps mis** 2 min 17 s

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

Question **1**

Correct

Note de 1,00 sur 1,00

Quelle est la bonne signature de l'opérateur d'affectation pour la classe **Compagnie**?

Veuillez choisir une réponse.

- ☐ a. `Compagnie operator=(Compagnie c);`
- ☒ b. `Compagnie& operator=(const Compagnie& c);` ✓
- ☐ c. `void operator=(const Compagnie& c);`
- ☐ d. `Compagnie& operator=(Compagnie c);`
- ☐ e. `void operator=();`
- ☐ f. `Compagnie operator=(const Compagnie& c);`
- ☐ g. `Compagnie* operator=(const Compagnie& c);`
- ☐ h. `Compagnie* operator=(const Compagnie* c);`

Votre réponse est correcte.

La réponse correcte est : `Compagnie& operator=(const Compagnie& c);`

## Question 2

Correct

Note de 1,00 sur 1,00

Soit le code suivant:

```
class Pilote {};  
class Passager {};  
  
class Vol {  
public:  
    Vol(Pilote& pilote, string id = "") : pilote_(pilote), id_(id) {}  
  
    void ajouterPassager(shared_ptr<Passager> passager) {  
        passagers_.push_back(passager);  
    }  
    ~Vol() {}  
  
private:  
    Pilote& pilote_;  
    vector< shared_ptr<Passager>> passagers_;  
    string id_;  
};
```

Quel(s) énoncé(s) est/sont vrai(s)?

Veuillez choisir au moins une réponse.

- ☐ 1. Il est présentement possible d'appeler la fonction ajouterPassager en cascade.
- ☐ 2. Le constructeur de copie de la classe Pilote sera appelé lorsqu'un object Vol est construit.
- ☐ 3. L'opérateur de copie devra absolument être implémenté pour la classe Vol pour pouvoir créer un nouveau Vol par copie.
- ☒ 4. Il est inutile de surcharger l'opérateur d'affectation pour la classe Vol. ✓

Votre réponse est correcte.

La réponse correcte est : Il est inutile de surcharger l'opérateur d'affectation pour la classe Vol.

## Question 3

Correct

Note de 2,00 sur 2,00

Soit une class **A** ayant un constructeur par défaut, un constructeur de copie et un operator= .

Pour chaque ligne du programme, indiquez s'il y a un appel au constructeur de copie. Si un passage de paramètre a besoin d'une copie, indiquez-la copie sur l'entête de fonction; si un retour fait une copie, indiquez-la sur le return.

Les sous-questions sont des lignes qui se suivent d'un même programme.

void f1(const A& x) {	jamais	✓
A y = x;		
}	toujours	✓
A f2(A x) {	toujours	✓
A y;	jamais	✓
y = x;	jamais	✓
return y;		
}	il peut ou non y avoir une copie	✓

Votre réponse est correcte.

La réponse correcte est : void f1(const A& x) { → jamais,   A y = x;  
} → toujours, A f2(A x) { → toujours,   A y; → jamais,   y = x; → jamais,   return y;  
} → il peut ou non y avoir une copie

## Question 4

Correct

Note de 1,00 sur 1,00

Sélectionner les éléments décrivant une fonction friend.

Veillez choisir au moins une réponse.

- ☐ 1. Est une fonction membre
- ☒ 2. A accès aux éléments privés d'une classe. ✓
- ☐ 3. A seulement accès aux fonctions membres d'une classe, qu'elles soient publiques ou privées
- ☒ 4. Est une fonction globale ✓

Votre réponse est correcte.

Les réponses correctes sont : A accès aux éléments privés d'une classe., Est une fonction globale

## Question 5

Correct

Note de 1,00 sur 1,00

Il n'est pas possible de surcharger les opérateurs pour les types primitifs tel que int, float ou char.

Veuillez choisir une réponse.

- ☒ Vrai ✓
- ☐ Faux

La réponse correcte est « Vrai ».

## Question 6

Correct

Note de 1,00 sur 1,00

Soit le code suivant:

```
int main() {  
    Employe e1("Marc", 150000);  
    Employe e2 = e1;  
}
```

Vrai ou faux. Il y a exactement un appel au constructeur de copie de la classe Employe et un appel à l'opérateur d'affectation (=) qui sont faits lors de l'exécution du code.

Veuillez choisir une réponse.

- ☐ Vrai
- ☒ Faux ✓

La réponse correcte est « Faux ».

## Question 7

Correct

Note de 2,00 sur 2,00

Soit le code ci-bas.

Écrivez l'implémentation de la fonction *incrémenter* qui augmente d'une unité les coordonnées (en x et en y) d'une instance de la classe *Point* et qui peut être appelée en cascade.

```
class Point
{
public:
    Point(int x, int y)
    {
        x_ = x;
        y_ = y;
    }

    void afficher() const
    {
        cout << "(" << x_ << ", " << y_ << ")" << endl;
    }

    Point& ☒ incrémenter( ☒ ) ☒
    {
        x_++;
        y_++;
        ☒
    }

private:
    int x_, y_;
};

// La fonction incrémenter sera implémentée ici.

int main()
{
    Point P(3,4);
    P.incrémenter().incrémenter().incrémenter();
    P.afficher();
}

// La sortie du programme est:
// (6,7)
```

Votre réponse est correcte.

La réponse correcte est :

Soit le code ci-bas.

Écrivez l'implémentation de la fonction *incrémenter* qui augmente d'une unité les coordonnées (en x et en y) d'une instance de la classe *Point* et qui peut être appelée en cascade.

```
class Point
{
public:
    Point(int x, int y)
    {
        x_ = x;
        y_ = y;
    }

    void afficher() const
    {
        cout << "(" << x_ << "," << y_ << ")" << endl;
    }

    [Point&] incrementer([[rien]]) [[rien]]
    {
        x_++;
        y_++;
        [return *this]
    }

private:
    int x_, y_;
};

// La fonction incrementer sera implémentée ici.

int main()
{
    Point P(3,4);
    P.incrementer().incrementer().incrementer();
    P.afficher();
}

// La sortie du programme est:
// (6,7)
```

## Question 8

Correct

Note de 1,00 sur 1,00

Pour une classe, si on a surchargé l'opérateur "+", implicitement, l'opérateur "+=" est aussi surchargé.

Veuillez choisir une réponse.

- ☐ Vrai
- ☒ Faux ✓

La réponse correcte est « Faux ».

[◀ Quiz 2-Hiver2023](#)

Aller à...

[Quiz 4-Hiver2023 ▶](#)