



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- procédural
- fonctionnel
- à objets

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre effectif transmis par valeur
- un paramètre formel transmis par référence
- un paramètre formel transmis par valeur
- un paramètre effectif transmis par référence

Question 3 En C++, la déclaration d'une classe :

- peut contenir 0 ou plusieurs constructeurs
- doit toujours contenir 2 constructeurs
- ne doit contenir qu'1 seul constructeur
- doit contenir au moins 1 constructeur

Question 4 En C++, on distingue des méthodes surchargées en fonction :

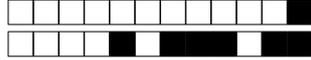
- de leurs noms
- uniquement du nombre de leurs paramètres
- de leurs types de retour
- du nombre ou du type de leurs paramètres

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this=this-c`
- `this+(-c)`
- `this-c`
- `this-*c`

Question 6 En C++, l'opérateur `new` permet de :

- créer un objet alloué dynamiquement
- définir un nouveau type alloué dynamiquement
- définir une nouvelle classe
- d'initialiser un nouveau type



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- le constructeur de copie de la classe `C`
- la surcharge de l'opérateur `=` de la classe `C`
- le destructeur la classe `C`
- rien du tout

Question 8 En C++, l'instruction `delete [] p;` :

- supprime le premier élément du tableau `p`
- déclare un tableau `p` de type `delete`
- supprime les éléments du tableau `p` alloués dynamiquement
- supprime la variable `p`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque l'appel du constructeur de copie de `C`
- provoque une erreur de compilation
- provoque une erreur à l'exécution
- ne provoque pas l'appel du constructeur de copie de `C`
- provoque l'appel du destructeur `C`

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge les opérateurs `<` et `<<`
- surcharge uniquement l'opérateur `<`
- surcharge uniquement l'opérateur `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- procédural
- fonctionnel
- à objets

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre effectif transmis par référence
- un paramètre formel transmis par valeur
- un paramètre formel transmis par référence
- un paramètre effectif transmis par valeur

Question 3 En C++, la déclaration d'une classe :

- peut contenir 0 ou plusieurs constructeurs
- doit contenir au moins 1 constructeur
- ne doit contenir qu'1 seul constructeur
- doit toujours contenir 2 constructeurs

Question 4 En C++, on distingue des méthodes surchargées en fonction :

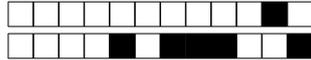
- de leurs noms
- de leurs types de retour
- uniquement du nombre de leurs paramètres
- du nombre ou du type de leurs paramètres

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this-c`
- `this+(-c)`
- `this=this-c`
- `this-*c`

Question 6 En C++, l'opérateur `new` permet de :

- créer un objet alloué dynamiquement
- définir une nouvelle classe
- définir un nouveau type alloué dynamiquement
- d'initialiser un nouveau type



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- le destructeur la classe `C`
- rien du tout
- le constructeur de copie de la classe `C`
- la surcharge de l'opérateur `=` de la classe `C`

Question 8 En C++, l'instruction `delete [] p;` :

- supprime les éléments du tableau `p` alloués dynamiquement
- supprime la variable `p`
- supprime le premier élément du tableau `p`
- déclare un tableau `p` de type `delete`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque l'appel du destructeur `C`
- provoque une erreur à l'exécution
- provoque une erreur de compilation
- provoque l'appel du constructeur de copie de `C`
- ne provoque pas l'appel du constructeur de copie de `C`

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge uniquement l'opérateur `<`
- surcharge uniquement l'opérateur `<<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge les opérateurs `<` et `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- à objets
- procédural
- fonctionnel

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre formel transmis par valeur
- un paramètre effectif transmis par référence
- un paramètre effectif transmis par valeur
- un paramètre formel transmis par référence

Question 3 En C++, la déclaration d'une classe :

- ne doit contenir qu'1 seul constructeur
- doit contenir au moins 1 constructeur
- doit toujours contenir 2 constructeurs
- peut contenir 0 ou plusieurs constructeurs

Question 4 En C++, on distingue des méthodes surchargées en fonction :

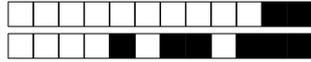
- uniquement du nombre de leurs paramètres
- de leurs types de retour
- du nombre ou du type de leurs paramètres
- de leurs noms

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this-c`
- `this+(-c)`
- `this-*c`
- `this=this-c`

Question 6 En C++, l'opérateur `new` permet de :

- définir une nouvelle classe
- créer un objet alloué dynamiquement
- définir un nouveau type alloué dynamiquement
- d'initialiser un nouveau type



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- le destructeur la classe `C`
- la surcharge de l'opérateur `=` de la classe `C`
- rien du tout
- le constructeur de copie de la classe `C`

Question 8 En C++, l'instruction `delete [] p;` :

- déclare un tableau `p` de type `delete`
- supprime la variable `p`
- supprime le premier élément du tableau `p`
- supprime les éléments du tableau `p` alloués dynamiquement

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque une erreur de compilation
- ne provoque pas l'appel du constructeur de copie de `C`
- provoque l'appel du constructeur de copie de `C`
- provoque une erreur à l'exécution
- provoque l'appel du destructeur `C`

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge uniquement l'opérateur `<<`
- surcharge les opérateurs `<` et `<<`
- surcharge uniquement l'opérateur `<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- procédural
- fonctionnel
- à objets

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre effectif transmis par valeur
- un paramètre effectif transmis par référence
- un paramètre formel transmis par référence
- un paramètre formel transmis par valeur

Question 3 En C++, la déclaration d'une classe :

- doit toujours contenir 2 constructeurs
- ne doit contenir qu'1 seul constructeur
- doit contenir au moins 1 constructeur
- peut contenir 0 ou plusieurs constructeurs

Question 4 En C++, on distingue des méthodes surchargées en fonction :

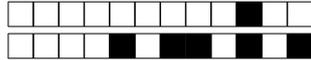
- uniquement du nombre de leurs paramètres
- de leurs noms
- de leurs types de retour
- du nombre ou du type de leurs paramètres

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this+(-c)`
- `this-*c`
- `this=this-c`
- `this-c`

Question 6 En C++, l'opérateur `new` permet de :

- définir un nouveau type alloué dynamiquement
- définir une nouvelle classe
- créer un objet alloué dynamiquement
- d'initialiser un nouveau type



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- le constructeur de copie de la classe `C`
- rien du tout
- le destructeur la classe `C`
- la surcharge de l'opérateur `=` de la classe `C`

Question 8 En C++, l'instruction `delete [] p;` :

- supprime le premier élément du tableau `p`
- supprime les éléments du tableau `p` alloués dynamiquement
- supprime la variable `p`
- déclare un tableau `p` de type `delete`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque l'appel du constructeur de copie de `C`
- provoque une erreur de compilation
- ne provoque pas l'appel du constructeur de copie de `C`
- provoque l'appel du destructeur `C`
- provoque une erreur à l'exécution

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge les opérateurs `<` et `<<`
- surcharge uniquement l'opérateur `<`
- surcharge uniquement l'opérateur `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- fonctionnel
- procédural
- à objets

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre formel transmis par référence
- un paramètre formel transmis par valeur
- un paramètre effectif transmis par valeur
- un paramètre effectif transmis par référence

Question 3 En C++, la déclaration d'une classe :

- doit contenir au moins 1 constructeur
- peut contenir 0 ou plusieurs constructeurs
- ne doit contenir qu'1 seul constructeur
- doit toujours contenir 2 constructeurs

Question 4 En C++, on distingue des méthodes surchargées en fonction :

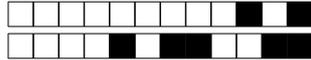
- de leurs noms
- de leurs types de retour
- uniquement du nombre de leurs paramètres
- du nombre ou du type de leurs paramètres

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this=this-c`
- `this+(-c)`
- `this-c`
- `this-*c`

Question 6 En C++, l'opérateur `new` permet de :

- définir un nouveau type alloué dynamiquement
- créer un objet alloué dynamiquement
- définir une nouvelle classe
- d'initialiser un nouveau type



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- rien du tout
- le destructeur la classe `C`
- la surcharge de l'opérateur `=` de la classe `C`
- le constructeur de copie de la classe `C`

Question 8 En C++, l'instruction `delete [] p;` :

- supprime le premier élément du tableau `p`
- supprime les éléments du tableau `p` alloués dynamiquement
- supprime la variable `p`
- déclare un tableau `p` de type `delete`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque l'appel du destructeur `C`
- provoque une erreur à l'exécution
- provoque une erreur de compilation
- provoque l'appel du constructeur de copie de `C`
- ne provoque pas l'appel du constructeur de copie de `C`

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge uniquement l'opérateur `<<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge uniquement l'opérateur `<`
- surcharge les opérateurs `<` et `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- procédural
- fonctionnel
- à objets

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre formel transmis par valeur
- un paramètre effectif transmis par référence
- un paramètre effectif transmis par valeur
- un paramètre formel transmis par référence

Question 3 En C++, la déclaration d'une classe :

- ne doit contenir qu'1 seul constructeur
- doit contenir au moins 1 constructeur
- doit toujours contenir 2 constructeurs
- peut contenir 0 ou plusieurs constructeurs

Question 4 En C++, on distingue des méthodes surchargées en fonction :

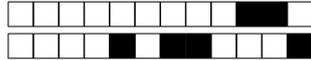
- du nombre ou du type de leurs paramètres
- de leurs noms
- uniquement du nombre de leurs paramètres
- de leurs types de retour

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this-c`
- `this-*c`
- `this+(-c)`
- `this=this-c`

Question 6 En C++, l'opérateur `new` permet de :

- définir une nouvelle classe
- créer un objet alloué dynamiquement
- définir un nouveau type alloué dynamiquement
- d'initialiser un nouveau type



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- la surcharge de l'opérateur = de la classe `C`
- le constructeur de copie de la classe `C`
- le destructeur la classe `C`
- rien du tout

Question 8 En C++, l'instruction `delete [] p;` :

- supprime la variable `p`
- supprime les éléments du tableau `p` alloués dynamiquement
- supprime le premier élément du tableau `p`
- déclare un tableau `p` de type `delete`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque l'appel du constructeur de copie de `C`
- ne provoque pas l'appel du constructeur de copie de `C`
- provoque une erreur à l'exécution
- provoque une erreur de compilation
- provoque l'appel du destructeur `C`

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge uniquement l'opérateur `<`
- surcharge uniquement l'opérateur `<<`
- surcharge les opérateurs `<` et `<<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- à objets
- procédural
- fonctionnel

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre formel transmis par valeur
- un paramètre effectif transmis par valeur
- un paramètre effectif transmis par référence
- un paramètre formel transmis par référence

Question 3 En C++, la déclaration d'une classe :

- doit toujours contenir 2 constructeurs
- doit contenir au moins 1 constructeur
- peut contenir 0 ou plusieurs constructeurs
- ne doit contenir qu'1 seul constructeur

Question 4 En C++, on distingue des méthodes surchargées en fonction :

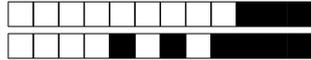
- uniquement du nombre de leurs paramètres
- de leurs noms
- du nombre ou du type de leurs paramètres
- de leurs types de retour

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this-c`
- `this=this-c`
- `this-*c`
- `this+(-c)`

Question 6 En C++, l'opérateur `new` permet de :

- définir une nouvelle classe
- définir un nouveau type alloué dynamiquement
- créer un objet alloué dynamiquement
- d'initialiser un nouveau type



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- la surcharge de l'opérateur = de la classe `C`
- le constructeur de copie de la classe `C`
- le destructeur la classe `C`
- rien du tout

Question 8 En C++, l'instruction `delete [] p;` :

- déclare un tableau `p` de type `delete`
- supprime les éléments du tableau `p` alloués dynamiquement
- supprime la variable `p`
- supprime le premier élément du tableau `p`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque une erreur de compilation
- provoque une erreur à l'exécution
- provoque l'appel du destructeur `C`
- ne provoque pas l'appel du constructeur de copie de `C`
- provoque l'appel du constructeur de copie de `C`

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge uniquement l'opérateur `<<`
- surcharge uniquement l'opérateur `<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge les opérateurs `<` et `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- procédural
- à objets
- fonctionnel

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre formel transmis par référence
- un paramètre formel transmis par valeur
- un paramètre effectif transmis par valeur
- un paramètre effectif transmis par référence

Question 3 En C++, la déclaration d'une classe :

- doit toujours contenir 2 constructeurs
- ne doit contenir qu'1 seul constructeur
- doit contenir au moins 1 constructeur
- peut contenir 0 ou plusieurs constructeurs

Question 4 En C++, on distingue des méthodes surchargées en fonction :

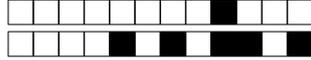
- de leurs noms
- uniquement du nombre de leurs paramètres
- de leurs types de retour
- du nombre ou du type de leurs paramètres

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this=this-c`
- `this-c`
- `this+(-c)`
- `this-*c`

Question 6 En C++, l'opérateur `new` permet de :

- définir une nouvelle classe
- d'initialiser un nouveau type
- créer un objet alloué dynamiquement
- définir un nouveau type alloué dynamiquement



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- la surcharge de l'opérateur = de la classe `C`
- rien du tout
- le destructeur la classe `C`
- le constructeur de copie de la classe `C`

Question 8 En C++, l'instruction `delete [] p;` :

- supprime le premier élément du tableau `p`
- supprime la variable `p`
- supprime les éléments du tableau `p` alloués dynamiquement
- déclare un tableau `p` de type `delete`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque l'appel du constructeur de copie de `C`
- provoque l'appel du destructeur `C`
- provoque une erreur à l'exécution
- ne provoque pas l'appel du constructeur de copie de `C`
- provoque une erreur de compilation

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge uniquement l'opérateur `<<`
- surcharge les opérateurs `<` et `<<`
- surcharge uniquement l'opérateur `<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- à objets
- fonctionnel
- procédural

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre effectif transmis par référence
- un paramètre formel transmis par référence
- un paramètre formel transmis par valeur
- un paramètre effectif transmis par valeur

Question 3 En C++, la déclaration d'une classe :

- doit contenir au moins 1 constructeur
- ne doit contenir qu'1 seul constructeur
- doit toujours contenir 2 constructeurs
- peut contenir 0 ou plusieurs constructeurs

Question 4 En C++, on distingue des méthodes surchargées en fonction :

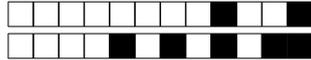
- de leurs types de retour
- uniquement du nombre de leurs paramètres
- de leurs noms
- du nombre ou du type de leurs paramètres

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this-*c`
- `this=this-c`
- `this+(-c)`
- `this-c`

Question 6 En C++, l'opérateur `new` permet de :

- créer un objet alloué dynamiquement
- définir une nouvelle classe
- d'initialiser un nouveau type
- définir un nouveau type alloué dynamiquement



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- le destructeur la classe `C`
- la surcharge de l'opérateur `=` de la classe `C`
- rien du tout
- le constructeur de copie de la classe `C`

Question 8 En C++, l'instruction `delete [] p;` :

- supprime la variable `p`
- déclare un tableau `p` de type `delete`
- supprime les éléments du tableau `p` alloués dynamiquement
- supprime le premier élément du tableau `p`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque une erreur de compilation
- ne provoque pas l'appel du constructeur de copie de `C`
- provoque l'appel du constructeur de copie de `C`
- provoque une erreur à l'exécution
- provoque l'appel du destructeur `C`

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge les opérateurs `<` et `<<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge uniquement l'opérateur `<<`
- surcharge uniquement l'opérateur `<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- à objets
- procédural
- fonctionnel

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre effectif transmis par valeur
- un paramètre formel transmis par valeur
- un paramètre effectif transmis par référence
- un paramètre formel transmis par référence

Question 3 En C++, la déclaration d'une classe :

- ne doit contenir qu'1 seul constructeur
- peut contenir 0 ou plusieurs constructeurs
- doit toujours contenir 2 constructeurs
- doit contenir au moins 1 constructeur

Question 4 En C++, on distingue des méthodes surchargées en fonction :

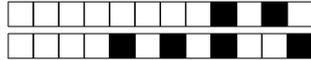
- uniquement du nombre de leurs paramètres
- de leurs noms
- du nombre ou du type de leurs paramètres
- de leurs types de retour

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this=this-c`
- `this-c`
- `this-*c`
- `this+(-c)`

Question 6 En C++, l'opérateur `new` permet de :

- définir une nouvelle classe
- d'initialiser un nouveau type
- définir un nouveau type alloué dynamiquement
- créer un objet alloué dynamiquement



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- le constructeur de copie de la classe `C`
- la surcharge de l'opérateur `=` de la classe `C`
- rien du tout
- le destructeur la classe `C`

Question 8 En C++, l'instruction `delete [] p;` :

- supprime les éléments du tableau `p` alloués dynamiquement
- supprime le premier élément du tableau `p`
- déclare un tableau `p` de type `delete`
- supprime la variable `p`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque une erreur de compilation
- provoque une erreur à l'exécution
- provoque l'appel du destructeur `C`
- provoque l'appel du constructeur de copie de `C`
- ne provoque pas l'appel du constructeur de copie de `C`

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge les opérateurs `<` et `<<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge uniquement l'opérateur `<`
- surcharge uniquement l'opérateur `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- fonctionnel
- procédural
- à objets

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre formel transmis par référence
- un paramètre effectif transmis par valeur
- un paramètre formel transmis par valeur
- un paramètre effectif transmis par référence

Question 3 En C++, la déclaration d'une classe :

- doit contenir au moins 1 constructeur
- doit toujours contenir 2 constructeurs
- ne doit contenir qu'1 seul constructeur
- peut contenir 0 ou plusieurs constructeurs

Question 4 En C++, on distingue des méthodes surchargées en fonction :

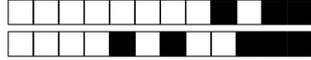
- de leurs types de retour
- de leurs noms
- du nombre ou du type de leurs paramètres
- uniquement du nombre de leurs paramètres

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this-c`
- `this-*c`
- `this=this-c`
- `this+(-c)`

Question 6 En C++, l'opérateur `new` permet de :

- d'initialiser un nouveau type
- définir un nouveau type alloué dynamiquement
- créer un objet alloué dynamiquement
- définir une nouvelle classe



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- le destructeur la classe `C`
- la surcharge de l'opérateur `=` de la classe `C`
- le constructeur de copie de la classe `C`
- rien du tout

Question 8 En C++, l'instruction `delete [] p;` :

- supprime la variable `p`
- supprime les éléments du tableau `p` alloués dynamiquement
- déclare un tableau `p` de type `delete`
- supprime le premier élément du tableau `p`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque une erreur à l'exécution
- provoque une erreur de compilation
- ne provoque pas l'appel du constructeur de copie de `C`
- provoque l'appel du destructeur `C`
- provoque l'appel du constructeur de copie de `C`

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge uniquement l'opérateur `<<`
- surcharge les opérateurs `<` et `<<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge uniquement l'opérateur `<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- procédural
- à objets
- fonctionnel

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre effectif transmis par valeur
- un paramètre formel transmis par référence
- un paramètre effectif transmis par référence
- un paramètre formel transmis par valeur

Question 3 En C++, la déclaration d'une classe :

- doit contenir au moins 1 constructeur
- peut contenir 0 ou plusieurs constructeurs
- ne doit contenir qu'1 seul constructeur
- doit toujours contenir 2 constructeurs

Question 4 En C++, on distingue des méthodes surchargées en fonction :

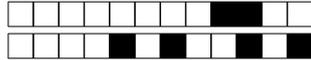
- du nombre ou du type de leurs paramètres
- uniquement du nombre de leurs paramètres
- de leurs noms
- de leurs types de retour

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this+(-c)`
- `this-c`
- `this=this-c`
- `this-*c`

Question 6 En C++, l'opérateur `new` permet de :

- créer un objet alloué dynamiquement
- d'initialiser un nouveau type
- définir une nouvelle classe
- définir un nouveau type alloué dynamiquement



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- la surcharge de l'opérateur = de la classe C
- rien du tout
- le destructeur la classe C
- le constructeur de copie de la classe C

Question 8 En C++, l'instruction `delete [] p;` :

- supprime le premier élément du tableau p
- supprime les éléments du tableau p alloués dynamiquement
- déclare un tableau p de type `delete`
- supprime la variable p

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

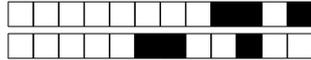
- provoque l'appel du destructeur C
- ne provoque pas l'appel du constructeur de copie de C
- provoque une erreur à l'exécution
- provoque une erreur de compilation
- provoque l'appel du constructeur de copie de C

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- ne surcharge *ni* l'opérateur <, *ni* l'opérateur <<
- surcharge uniquement l'opérateur <
- surcharge uniquement l'opérateur <<
- surcharge les opérateurs < et <<



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- à objets
- procédural
- fonctionnel

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre formel transmis par valeur
- un paramètre effectif transmis par référence
- un paramètre effectif transmis par valeur
- un paramètre formel transmis par référence

Question 3 En C++, la déclaration d'une classe :

- doit toujours contenir 2 constructeurs
- peut contenir 0 ou plusieurs constructeurs
- doit contenir au moins 1 constructeur
- ne doit contenir qu'1 seul constructeur

Question 4 En C++, on distingue des méthodes surchargées en fonction :

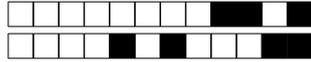
- du nombre ou du type de leurs paramètres
- uniquement du nombre de leurs paramètres
- de leurs noms
- de leurs types de retour

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this=this-c`
- `this-c`
- `this-*c`
- `this+(-c)`

Question 6 En C++, l'opérateur `new` permet de :

- d'initialiser un nouveau type
- définir une nouvelle classe
- définir un nouveau type alloué dynamiquement
- créer un objet alloué dynamiquement



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- le constructeur de copie de la classe `C`
- la surcharge de l'opérateur `=` de la classe `C`
- le destructeur la classe `C`
- rien du tout

Question 8 En C++, l'instruction `delete [] p;` :

- supprime la variable `p`
- déclare un tableau `p` de type `delete`
- supprime le premier élément du tableau `p`
- supprime les éléments du tableau `p` alloués dynamiquement

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- ne provoque pas l'appel du constructeur de copie de `C`
- provoque l'appel du destructeur `C`
- provoque une erreur à l'exécution
- provoque l'appel du constructeur de copie de `C`
- provoque une erreur de compilation

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge uniquement l'opérateur `<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge uniquement l'opérateur `<<`
- surcharge les opérateurs `<` et `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- fonctionnel
- procédural
- à objets

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre effectif transmis par valeur
- un paramètre effectif transmis par référence
- un paramètre formel transmis par référence
- un paramètre formel transmis par valeur

Question 3 En C++, la déclaration d'une classe :

- peut contenir 0 ou plusieurs constructeurs
- doit toujours contenir 2 constructeurs
- doit contenir au moins 1 constructeur
- ne doit contenir qu'1 seul constructeur

Question 4 En C++, on distingue des méthodes surchargées en fonction :

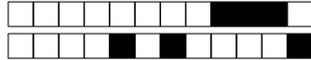
- de leurs noms
- du nombre ou du type de leurs paramètres
- de leurs types de retour
- uniquement du nombre de leurs paramètres

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this+(-c)`
- `this-*c`
- `this-c`
- `this=this-c`

Question 6 En C++, l'opérateur `new` permet de :

- d'initialiser un nouveau type
- définir un nouveau type alloué dynamiquement
- définir une nouvelle classe
- créer un objet alloué dynamiquement



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- le constructeur de copie de la classe `C`
- le destructeur la classe `C`
- rien du tout
- la surcharge de l'opérateur `=` de la classe `C`

Question 8 En C++, l'instruction `delete [] p;` :

- déclare un tableau `p` de type `delete`
- supprime la variable `p`
- supprime les éléments du tableau `p` alloués dynamiquement
- supprime le premier élément du tableau `p`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque une erreur à l'exécution
- provoque l'appel du destructeur `C`
- provoque une erreur de compilation
- provoque l'appel du constructeur de copie de `C`
- ne provoque pas l'appel du constructeur de copie de `C`

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge uniquement l'opérateur `<`
- surcharge les opérateurs `<` et `<<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge uniquement l'opérateur `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- à objets
- fonctionnel
- procédural

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre effectif transmis par référence
- un paramètre effectif transmis par valeur
- un paramètre formel transmis par valeur
- un paramètre formel transmis par référence

Question 3 En C++, la déclaration d'une classe :

- peut contenir 0 ou plusieurs constructeurs
- ne doit contenir qu'1 seul constructeur
- doit contenir au moins 1 constructeur
- doit toujours contenir 2 constructeurs

Question 4 En C++, on distingue des méthodes surchargées en fonction :

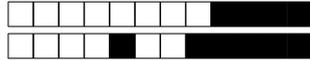
- uniquement du nombre de leurs paramètres
- du nombre ou du type de leurs paramètres
- de leurs types de retour
- de leurs noms

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this=this-c`
- `this-c`
- `this-*c`
- `this+(-c)`

Question 6 En C++, l'opérateur `new` permet de :

- définir un nouveau type alloué dynamiquement
- créer un objet alloué dynamiquement
- définir une nouvelle classe
- d'initialiser un nouveau type



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- le destructeur la classe `C`
- le constructeur de copie de la classe `C`
- la surcharge de l'opérateur `=` de la classe `C`
- rien du tout

Question 8 En C++, l'instruction `delete [] p;` :

- déclare un tableau `p` de type `delete`
- supprime la variable `p`
- supprime les éléments du tableau `p` alloués dynamiquement
- supprime le premier élément du tableau `p`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque l'appel du constructeur de copie de `C`
- ne provoque pas l'appel du constructeur de copie de `C`
- provoque une erreur de compilation
- provoque l'appel du destructeur `C`
- provoque une erreur à l'exécution

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge les opérateurs `<` et `<<`
- surcharge uniquement l'opérateur `<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge uniquement l'opérateur `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- fonctionnel
- à objets
- procédural

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre formel transmis par référence
- un paramètre effectif transmis par référence
- un paramètre formel transmis par valeur
- un paramètre effectif transmis par valeur

Question 3 En C++, la déclaration d'une classe :

- doit contenir au moins 1 constructeur
- doit toujours contenir 2 constructeurs
- ne doit contenir qu'1 seul constructeur
- peut contenir 0 ou plusieurs constructeurs

Question 4 En C++, on distingue des méthodes surchargées en fonction :

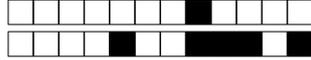
- de leurs noms
- du nombre ou du type de leurs paramètres
- uniquement du nombre de leurs paramètres
- de leurs types de retour

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this-*c`
- `this-c`
- `this+(-c)`
- `this=this-c`

Question 6 En C++, l'opérateur `new` permet de :

- créer un objet alloué dynamiquement
- d'initialiser un nouveau type
- définir une nouvelle classe
- définir un nouveau type alloué dynamiquement



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- le destructeur la classe `C`
- rien du tout
- le constructeur de copie de la classe `C`
- la surcharge de l'opérateur `=` de la classe `C`

Question 8 En C++, l'instruction `delete [] p;` :

- déclare un tableau `p` de type `delete`
- supprime la variable `p`
- supprime les éléments du tableau `p` alloués dynamiquement
- supprime le premier élément du tableau `p`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque l'appel du destructeur `C`
- ne provoque pas l'appel du constructeur de copie de `C`
- provoque une erreur de compilation
- provoque l'appel du constructeur de copie de `C`
- provoque une erreur à l'exécution

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge uniquement l'opérateur `<<`
- surcharge les opérateurs `<` et `<<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge uniquement l'opérateur `<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- à objets
- fonctionnel
- procédural

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre formel transmis par valeur
- un paramètre effectif transmis par valeur
- un paramètre formel transmis par référence
- un paramètre effectif transmis par référence

Question 3 En C++, la déclaration d'une classe :

- ne doit contenir qu'1 seul constructeur
- peut contenir 0 ou plusieurs constructeurs
- doit toujours contenir 2 constructeurs
- doit contenir au moins 1 constructeur

Question 4 En C++, on distingue des méthodes surchargées en fonction :

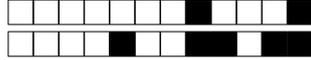
- de leurs noms
- uniquement du nombre de leurs paramètres
- de leurs types de retour
- du nombre ou du type de leurs paramètres

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this-*c`
- `this+(-c)`
- `this=this-c`
- `this-c`

Question 6 En C++, l'opérateur `new` permet de :

- définir un nouveau type alloué dynamiquement
- définir une nouvelle classe
- créer un objet alloué dynamiquement
- d'initialiser un nouveau type



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- le destructeur la classe `C`
- le constructeur de copie de la classe `C`
- rien du tout
- la surcharge de l'opérateur `=` de la classe `C`

Question 8 En C++, l'instruction `delete [] p;` :

- supprime le premier élément du tableau `p`
- déclare un tableau `p` de type `delete`
- supprime la variable `p`
- supprime les éléments du tableau `p` alloués dynamiquement

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque une erreur à l'exécution
- provoque l'appel du constructeur de copie de `C`
- provoque une erreur de compilation
- provoque l'appel du destructeur `C`
- ne provoque pas l'appel du constructeur de copie de `C`

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge uniquement l'opérateur `<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge les opérateurs `<` et `<<`
- surcharge uniquement l'opérateur `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- procédural
- fonctionnel
- à objets

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre effectif transmis par valeur
- un paramètre formel transmis par valeur
- un paramètre formel transmis par référence
- un paramètre effectif transmis par référence

Question 3 En C++, la déclaration d'une classe :

- peut contenir 0 ou plusieurs constructeurs
- ne doit contenir qu'1 seul constructeur
- doit toujours contenir 2 constructeurs
- doit contenir au moins 1 constructeur

Question 4 En C++, on distingue des méthodes surchargées en fonction :

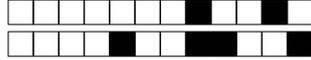
- de leurs noms
- uniquement du nombre de leurs paramètres
- de leurs types de retour
- du nombre ou du type de leurs paramètres

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this+(-c)`
- `this-*c`
- `this=this-c`
- `this-c`

Question 6 En C++, l'opérateur `new` permet de :

- définir une nouvelle classe
- définir un nouveau type alloué dynamiquement
- créer un objet alloué dynamiquement
- d'initialiser un nouveau type



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- le constructeur de copie de la classe `C`
- la surcharge de l'opérateur `=` de la classe `C`
- le destructeur la classe `C`
- rien du tout

Question 8 En C++, l'instruction `delete [] p;` :

- supprime la variable `p`
- déclare un tableau `p` de type `delete`
- supprime le premier élément du tableau `p`
- supprime les éléments du tableau `p` alloués dynamiquement

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque l'appel du destructeur `C`
- provoque une erreur à l'exécution
- ne provoque pas l'appel du constructeur de copie de `C`
- provoque l'appel du constructeur de copie de `C`
- provoque une erreur de compilation

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge uniquement l'opérateur `<<`
- surcharge les opérateurs `<` et `<<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge uniquement l'opérateur `<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- fonctionnel
- procédural
- à objets

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre effectif transmis par référence
- un paramètre formel transmis par référence
- un paramètre formel transmis par valeur
- un paramètre effectif transmis par valeur

Question 3 En C++, la déclaration d'une classe :

- peut contenir 0 ou plusieurs constructeurs
- ne doit contenir qu'1 seul constructeur
- doit contenir au moins 1 constructeur
- doit toujours contenir 2 constructeurs

Question 4 En C++, on distingue des méthodes surchargées en fonction :

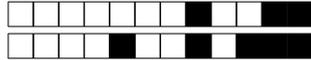
- uniquement du nombre de leurs paramètres
- de leurs noms
- du nombre ou du type de leurs paramètres
- de leurs types de retour

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this+(-c)`
- `this-c`
- `this-*c`
- `this=this-c`

Question 6 En C++, l'opérateur `new` permet de :

- créer un objet alloué dynamiquement
- définir une nouvelle classe
- définir un nouveau type alloué dynamiquement
- d'initialiser un nouveau type



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- le destructeur la classe `C`
- rien du tout
- le constructeur de copie de la classe `C`
- la surcharge de l'opérateur `=` de la classe `C`

Question 8 En C++, l'instruction `delete [] p;` :

- supprime le premier élément du tableau `p`
- déclare un tableau `p` de type `delete`
- supprime les éléments du tableau `p` alloués dynamiquement
- supprime la variable `p`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque l'appel du destructeur `C`
- provoque l'appel du constructeur de copie de `C`
- provoque une erreur à l'exécution
- provoque une erreur de compilation
- ne provoque pas l'appel du constructeur de copie de `C`

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge uniquement l'opérateur `<`
- surcharge les opérateurs `<` et `<<`
- surcharge uniquement l'opérateur `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- à objets
- procédural
- fonctionnel

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre effectif transmis par référence
- un paramètre formel transmis par valeur
- un paramètre effectif transmis par valeur
- un paramètre formel transmis par référence

Question 3 En C++, la déclaration d'une classe :

- peut contenir 0 ou plusieurs constructeurs
- doit toujours contenir 2 constructeurs
- ne doit contenir qu'1 seul constructeur
- doit contenir au moins 1 constructeur

Question 4 En C++, on distingue des méthodes surchargées en fonction :

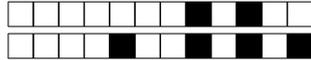
- du nombre ou du type de leurs paramètres
- de leurs noms
- uniquement du nombre de leurs paramètres
- de leurs types de retour

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this=this-c`
- `this-c`
- `this-*c`
- `this+(-c)`

Question 6 En C++, l'opérateur `new` permet de :

- d'initialiser un nouveau type
- définir une nouvelle classe
- définir un nouveau type alloué dynamiquement
- créer un objet alloué dynamiquement



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- le destructeur la classe `C`
- la surcharge de l'opérateur `=` de la classe `C`
- le constructeur de copie de la classe `C`
- rien du tout

Question 8 En C++, l'instruction `delete [] p;` :

- supprime le premier élément du tableau `p`
- supprime la variable `p`
- supprime les éléments du tableau `p` alloués dynamiquement
- déclare un tableau `p` de type `delete`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque l'appel du constructeur de copie de `C`
- provoque une erreur à l'exécution
- provoque une erreur de compilation
- provoque l'appel du destructeur `C`
- ne provoque pas l'appel du constructeur de copie de `C`

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge uniquement l'opérateur `<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge uniquement l'opérateur `<<`
- surcharge les opérateurs `<` et `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- fonctionnel
- procédural
- à objets

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre formel transmis par valeur
- un paramètre effectif transmis par référence
- un paramètre effectif transmis par valeur
- un paramètre formel transmis par référence

Question 3 En C++, la déclaration d'une classe :

- doit contenir au moins 1 constructeur
- ne doit contenir qu'1 seul constructeur
- doit toujours contenir 2 constructeurs
- peut contenir 0 ou plusieurs constructeurs

Question 4 En C++, on distingue des méthodes surchargées en fonction :

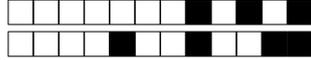
- du nombre ou du type de leurs paramètres
- uniquement du nombre de leurs paramètres
- de leurs types de retour
- de leurs noms

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this=this-c`
- `this+(-c)`
- `this-c`
- `this-*c`

Question 6 En C++, l'opérateur `new` permet de :

- définir un nouveau type alloué dynamiquement
- définir une nouvelle classe
- créer un objet alloué dynamiquement
- d'initialiser un nouveau type



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- le destructeur la classe `C`
- le constructeur de copie de la classe `C`
- la surcharge de l'opérateur `=` de la classe `C`
- rien du tout

Question 8 En C++, l'instruction `delete [] p;` :

- déclare un tableau `p` de type `delete`
- supprime la variable `p`
- supprime les éléments du tableau `p` alloués dynamiquement
- supprime le premier élément du tableau `p`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- ne provoque pas l'appel du constructeur de copie de `C`
- provoque l'appel du constructeur de copie de `C`
- provoque une erreur de compilation
- provoque l'appel du destructeur `C`
- provoque une erreur à l'exécution

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge uniquement l'opérateur `<<`
- surcharge les opérateurs `<` et `<<`
- surcharge uniquement l'opérateur `<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- fonctionnel
- à objets
- procédural

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre effectif transmis par référence
- un paramètre effectif transmis par valeur
- un paramètre formel transmis par valeur
- un paramètre formel transmis par référence

Question 3 En C++, la déclaration d'une classe :

- doit contenir au moins 1 constructeur
- doit toujours contenir 2 constructeurs
- peut contenir 0 ou plusieurs constructeurs
- ne doit contenir qu'1 seul constructeur

Question 4 En C++, on distingue des méthodes surchargées en fonction :

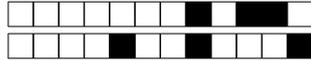
- uniquement du nombre de leurs paramètres
- de leurs noms
- de leurs types de retour
- du nombre ou du type de leurs paramètres

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this-c`
- `this=this-c`
- `this-*c`
- `this+(-c)`

Question 6 En C++, l'opérateur `new` permet de :

- créer un objet alloué dynamiquement
- définir un nouveau type alloué dynamiquement
- d'initialiser un nouveau type
- définir une nouvelle classe



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- rien du tout
- le destructeur la classe `C`
- le constructeur de copie de la classe `C`
- la surcharge de l'opérateur `=` de la classe `C`

Question 8 En C++, l'instruction `delete [] p;` :

- supprime le premier élément du tableau `p`
- supprime les éléments du tableau `p` alloués dynamiquement
- déclare un tableau `p` de type `delete`
- supprime la variable `p`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque l'appel du destructeur `C`
- provoque une erreur de compilation
- provoque une erreur à l'exécution
- provoque l'appel du constructeur de copie de `C`
- ne provoque pas l'appel du constructeur de copie de `C`

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge les opérateurs `<` et `<<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge uniquement l'opérateur `<<`
- surcharge uniquement l'opérateur `<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- procédural
- fonctionnel
- à objets

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre formel transmis par référence
- un paramètre formel transmis par valeur
- un paramètre effectif transmis par valeur
- un paramètre effectif transmis par référence

Question 3 En C++, la déclaration d'une classe :

- doit toujours contenir 2 constructeurs
- peut contenir 0 ou plusieurs constructeurs
- doit contenir au moins 1 constructeur
- ne doit contenir qu'1 seul constructeur

Question 4 En C++, on distingue des méthodes surchargées en fonction :

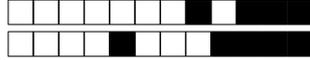
- de leurs types de retour
- de leurs noms
- du nombre ou du type de leurs paramètres
- uniquement du nombre de leurs paramètres

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this=this-c`
- `this-c`
- `this+(-c)`
- `this-*c`

Question 6 En C++, l'opérateur `new` permet de :

- d'initialiser un nouveau type
- définir une nouvelle classe
- définir un nouveau type alloué dynamiquement
- créer un objet alloué dynamiquement



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- le destructeur la classe `C`
- la surcharge de l'opérateur `=` de la classe `C`
- rien du tout
- le constructeur de copie de la classe `C`

Question 8 En C++, l'instruction `delete [] p;` :

- supprime les éléments du tableau `p` alloués dynamiquement
- supprime le premier élément du tableau `p`
- déclare un tableau `p` de type `delete`
- supprime la variable `p`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque l'appel du constructeur de copie de `C`
- provoque l'appel du destructeur `C`
- ne provoque pas l'appel du constructeur de copie de `C`
- provoque une erreur à l'exécution
- provoque une erreur de compilation

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge uniquement l'opérateur `<`
- surcharge uniquement l'opérateur `<<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge les opérateurs `<` et `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- procédural
- à objets
- fonctionnel

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre formel transmis par valeur
- un paramètre effectif transmis par référence
- un paramètre effectif transmis par valeur
- un paramètre formel transmis par référence

Question 3 En C++, la déclaration d'une classe :

- peut contenir 0 ou plusieurs constructeurs
- doit contenir au moins 1 constructeur
- doit toujours contenir 2 constructeurs
- ne doit contenir qu'1 seul constructeur

Question 4 En C++, on distingue des méthodes surchargées en fonction :

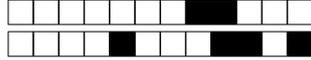
- de leurs noms
- de leurs types de retour
- uniquement du nombre de leurs paramètres
- du nombre ou du type de leurs paramètres

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this+(-c)`
- `this-c`
- `this=this-c`
- `this-*c`

Question 6 En C++, l'opérateur `new` permet de :

- définir une nouvelle classe
- d'initialiser un nouveau type
- créer un objet alloué dynamiquement
- définir un nouveau type alloué dynamiquement



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- la surcharge de l'opérateur `=` de la classe `C`
- rien du tout
- le destructeur la classe `C`
- le constructeur de copie de la classe `C`

Question 8 En C++, l'instruction `delete [] p;` :

- supprime le premier élément du tableau `p`
- supprime la variable `p`
- déclare un tableau `p` de type `delete`
- supprime les éléments du tableau `p` alloués dynamiquement

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- ne provoque pas l'appel du constructeur de copie de `C`
- provoque l'appel du destructeur `C`
- provoque une erreur de compilation
- provoque l'appel du constructeur de copie de `C`
- provoque une erreur à l'exécution

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge uniquement l'opérateur `<<`
- surcharge les opérateurs `<` et `<<`
- surcharge uniquement l'opérateur `<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- à objets
- fonctionnel
- procédural

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre effectif transmis par référence
- un paramètre formel transmis par valeur
- un paramètre effectif transmis par valeur
- un paramètre formel transmis par référence

Question 3 En C++, la déclaration d'une classe :

- ne doit contenir qu'1 seul constructeur
- doit toujours contenir 2 constructeurs
- peut contenir 0 ou plusieurs constructeurs
- doit contenir au moins 1 constructeur

Question 4 En C++, on distingue des méthodes surchargées en fonction :

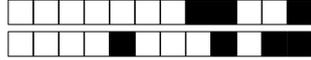
- de leurs noms
- du nombre ou du type de leurs paramètres
- de leurs types de retour
- uniquement du nombre de leurs paramètres

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this-*c`
- `this=this-c`
- `this-c`
- `this+(-c)`

Question 6 En C++, l'opérateur `new` permet de :

- définir un nouveau type alloué dynamiquement
- d'initialiser un nouveau type
- créer un objet alloué dynamiquement
- définir une nouvelle classe



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- le constructeur de copie de la classe `C`
- la surcharge de l'opérateur `=` de la classe `C`
- rien du tout
- le destructeur la classe `C`

Question 8 En C++, l'instruction `delete [] p;` :

- supprime le premier élément du tableau `p`
- supprime les éléments du tableau `p` alloués dynamiquement
- supprime la variable `p`
- déclare un tableau `p` de type `delete`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque une erreur à l'exécution
- provoque l'appel du constructeur de copie de `C`
- ne provoque pas l'appel du constructeur de copie de `C`
- provoque l'appel du destructeur `C`
- provoque une erreur de compilation

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge les opérateurs `<` et `<<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge uniquement l'opérateur `<`
- surcharge uniquement l'opérateur `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- fonctionnel
- procédural
- à objets

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre effectif transmis par référence
- un paramètre formel transmis par référence
- un paramètre formel transmis par valeur
- un paramètre effectif transmis par valeur

Question 3 En C++, la déclaration d'une classe :

- doit toujours contenir 2 constructeurs
- ne doit contenir qu'1 seul constructeur
- peut contenir 0 ou plusieurs constructeurs
- doit contenir au moins 1 constructeur

Question 4 En C++, on distingue des méthodes surchargées en fonction :

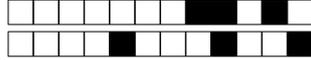
- de leurs noms
- du nombre ou du type de leurs paramètres
- uniquement du nombre de leurs paramètres
- de leurs types de retour

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this-*c`
- `this+(-c)`
- `this-c`
- `this=this-c`

Question 6 En C++, l'opérateur `new` permet de :

- définir une nouvelle classe
- créer un objet alloué dynamiquement
- d'initialiser un nouveau type
- définir un nouveau type alloué dynamiquement



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- rien du tout
- le destructeur la classe `C`
- la surcharge de l'opérateur `=` de la classe `C`
- le constructeur de copie de la classe `C`

Question 8 En C++, l'instruction `delete [] p;` :

- supprime le premier élément du tableau `p`
- déclare un tableau `p` de type `delete`
- supprime les éléments du tableau `p` alloués dynamiquement
- supprime la variable `p`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque l'appel du destructeur `C`
- provoque l'appel du constructeur de copie de `C`
- provoque une erreur à l'exécution
- ne provoque pas l'appel du constructeur de copie de `C`
- provoque une erreur de compilation

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge uniquement l'opérateur `<<`
- surcharge les opérateurs `<` et `<<`
- surcharge uniquement l'opérateur `<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- à objets
- fonctionnel
- procédural

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre effectif transmis par valeur
- un paramètre formel transmis par valeur
- un paramètre effectif transmis par référence
- un paramètre formel transmis par référence

Question 3 En C++, la déclaration d'une classe :

- doit contenir au moins 1 constructeur
- peut contenir 0 ou plusieurs constructeurs
- doit toujours contenir 2 constructeurs
- ne doit contenir qu'1 seul constructeur

Question 4 En C++, on distingue des méthodes surchargées en fonction :

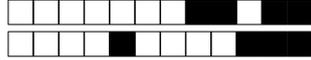
- de leurs noms
- de leurs types de retour
- uniquement du nombre de leurs paramètres
- du nombre ou du type de leurs paramètres

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this=this-c`
- `this-c`
- `this-*c`
- `this+(-c)`

Question 6 En C++, l'opérateur `new` permet de :

- définir un nouveau type alloué dynamiquement
- d'initialiser un nouveau type
- définir une nouvelle classe
- créer un objet alloué dynamiquement



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- la surcharge de l'opérateur = de la classe `C`
- le constructeur de copie de la classe `C`
- rien du tout
- le destructeur la classe `C`

Question 8 En C++, l'instruction `delete [] p;` :

- supprime les éléments du tableau `p` alloués dynamiquement
- déclare un tableau `p` de type `delete`
- supprime la variable `p`
- supprime le premier élément du tableau `p`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque une erreur de compilation
- provoque une erreur à l'exécution
- provoque l'appel du destructeur `C`
- ne provoque pas l'appel du constructeur de copie de `C`
- provoque l'appel du constructeur de copie de `C`

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge les opérateurs `<` et `<<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge uniquement l'opérateur `<`
- surcharge uniquement l'opérateur `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- procédural
- fonctionnel
- à objets

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre effectif transmis par valeur
- un paramètre formel transmis par référence
- un paramètre formel transmis par valeur
- un paramètre effectif transmis par référence

Question 3 En C++, la déclaration d'une classe :

- peut contenir 0 ou plusieurs constructeurs
- doit contenir au moins 1 constructeur
- doit toujours contenir 2 constructeurs
- ne doit contenir qu'1 seul constructeur

Question 4 En C++, on distingue des méthodes surchargées en fonction :

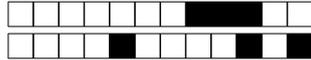
- uniquement du nombre de leurs paramètres
- de leurs types de retour
- du nombre ou du type de leurs paramètres
- de leurs noms

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this-c`
- `this-*c`
- `this=this-c`
- `this+(-c)`

Question 6 En C++, l'opérateur `new` permet de :

- définir une nouvelle classe
- d'initialiser un nouveau type
- créer un objet alloué dynamiquement
- définir un nouveau type alloué dynamiquement



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- le constructeur de copie de la classe `C`
- rien du tout
- la surcharge de l'opérateur `=` de la classe `C`
- le destructeur la classe `C`

Question 8 En C++, l'instruction `delete [] p;` :

- supprime les éléments du tableau `p` alloués dynamiquement
- supprime la variable `p`
- supprime le premier élément du tableau `p`
- déclare un tableau `p` de type `delete`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque l'appel du constructeur de copie de `C`
- provoque une erreur à l'exécution
- provoque l'appel du destructeur `C`
- provoque une erreur de compilation
- ne provoque pas l'appel du constructeur de copie de `C`

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge uniquement l'opérateur `<<`
- surcharge les opérateurs `<` et `<<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge uniquement l'opérateur `<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- à objets
- procédural
- fonctionnel

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre formel transmis par valeur
- un paramètre effectif transmis par valeur
- un paramètre effectif transmis par référence
- un paramètre formel transmis par référence

Question 3 En C++, la déclaration d'une classe :

- ne doit contenir qu'1 seul constructeur
- doit contenir au moins 1 constructeur
- peut contenir 0 ou plusieurs constructeurs
- doit toujours contenir 2 constructeurs

Question 4 En C++, on distingue des méthodes surchargées en fonction :

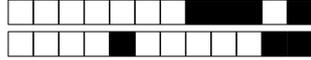
- de leurs types de retour
- uniquement du nombre de leurs paramètres
- du nombre ou du type de leurs paramètres
- de leurs noms

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this=this-c`
- `this+(-c)`
- `this-*c`
- `this-c`

Question 6 En C++, l'opérateur `new` permet de :

- créer un objet alloué dynamiquement
- définir une nouvelle classe
- d'initialiser un nouveau type
- définir un nouveau type alloué dynamiquement



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- le destructeur la classe `C`
- le constructeur de copie de la classe `C`
- la surcharge de l'opérateur `=` de la classe `C`
- rien du tout

Question 8 En C++, l'instruction `delete [] p;` :

- déclare un tableau `p` de type `delete`
- supprime la variable `p`
- supprime les éléments du tableau `p` alloués dynamiquement
- supprime le premier élément du tableau `p`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque l'appel du constructeur de copie de `C`
- provoque l'appel du destructeur `C`
- provoque une erreur à l'exécution
- provoque une erreur de compilation
- ne provoque pas l'appel du constructeur de copie de `C`

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge les opérateurs `<` et `<<`
- surcharge uniquement l'opérateur `<<`
- surcharge uniquement l'opérateur `<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- fonctionnel
- procédural
- à objets

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre formel transmis par valeur
- un paramètre formel transmis par référence
- un paramètre effectif transmis par valeur
- un paramètre effectif transmis par référence

Question 3 En C++, la déclaration d'une classe :

- doit contenir au moins 1 constructeur
- ne doit contenir qu'1 seul constructeur
- peut contenir 0 ou plusieurs constructeurs
- doit toujours contenir 2 constructeurs

Question 4 En C++, on distingue des méthodes surchargées en fonction :

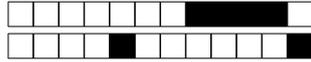
- de leurs noms
- du nombre ou du type de leurs paramètres
- uniquement du nombre de leurs paramètres
- de leurs types de retour

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this+(-c)`
- `this-*c`
- `this=this-c`
- `this-c`

Question 6 En C++, l'opérateur `new` permet de :

- créer un objet alloué dynamiquement
- d'initialiser un nouveau type
- définir un nouveau type alloué dynamiquement
- définir une nouvelle classe



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- le constructeur de copie de la classe `C`
- la surcharge de l'opérateur `=` de la classe `C`
- le destructeur la classe `C`
- rien du tout

Question 8 En C++, l'instruction `delete [] p;` :

- supprime les éléments du tableau `p` alloués dynamiquement
- supprime la variable `p`
- supprime le premier élément du tableau `p`
- déclare un tableau `p` de type `delete`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque l'appel du constructeur de copie de `C`
- provoque une erreur à l'exécution
- ne provoque pas l'appel du constructeur de copie de `C`
- provoque l'appel du destructeur `C`
- provoque une erreur de compilation

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge uniquement l'opérateur `<`
- surcharge les opérateurs `<` et `<<`
- surcharge uniquement l'opérateur `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- fonctionnel
- procédural
- à objets

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre effectif transmis par valeur
- un paramètre formel transmis par référence
- un paramètre effectif transmis par référence
- un paramètre formel transmis par valeur

Question 3 En C++, la déclaration d'une classe :

- peut contenir 0 ou plusieurs constructeurs
- ne doit contenir qu'1 seul constructeur
- doit toujours contenir 2 constructeurs
- doit contenir au moins 1 constructeur

Question 4 En C++, on distingue des méthodes surchargées en fonction :

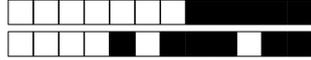
- uniquement du nombre de leurs paramètres
- du nombre ou du type de leurs paramètres
- de leurs types de retour
- de leurs noms

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this-c`
- `this=this-c`
- `this-*c`
- `this+(-c)`

Question 6 En C++, l'opérateur `new` permet de :

- définir une nouvelle classe
- d'initialiser un nouveau type
- créer un objet alloué dynamiquement
- définir un nouveau type alloué dynamiquement



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- la surcharge de l'opérateur = de la classe `C`
- le constructeur de copie de la classe `C`
- le destructeur la classe `C`
- rien du tout

Question 8 En C++, l'instruction `delete [] p;` :

- supprime les éléments du tableau `p` alloués dynamiquement
- déclare un tableau `p` de type `delete`
- supprime la variable `p`
- supprime le premier élément du tableau `p`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque l'appel du constructeur de copie de `C`
- provoque l'appel du destructeur `C`
- provoque une erreur de compilation
- provoque une erreur à l'exécution
- ne provoque pas l'appel du constructeur de copie de `C`

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge uniquement l'opérateur `<`
- surcharge les opérateurs `<` et `<<`
- surcharge uniquement l'opérateur `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- à objets
- fonctionnel
- procédural

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre effectif transmis par référence
- un paramètre formel transmis par valeur
- un paramètre formel transmis par référence
- un paramètre effectif transmis par valeur

Question 3 En C++, la déclaration d'une classe :

- doit toujours contenir 2 constructeurs
- ne doit contenir qu'1 seul constructeur
- doit contenir au moins 1 constructeur
- peut contenir 0 ou plusieurs constructeurs

Question 4 En C++, on distingue des méthodes surchargées en fonction :

- du nombre ou du type de leurs paramètres
- uniquement du nombre de leurs paramètres
- de leurs noms
- de leurs types de retour

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this-*c`
- `this+(-c)`
- `this-c`
- `this=this-c`

Question 6 En C++, l'opérateur `new` permet de :

- créer un objet alloué dynamiquement
- d'initialiser un nouveau type
- définir un nouveau type alloué dynamiquement
- définir une nouvelle classe



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- le constructeur de copie de la classe `C`
- le destructeur la classe `C`
- rien du tout
- la surcharge de l'opérateur `=` de la classe `C`

Question 8 En C++, l'instruction `delete [] p;` :

- supprime les éléments du tableau `p` alloués dynamiquement
- déclare un tableau `p` de type `delete`
- supprime la variable `p`
- supprime le premier élément du tableau `p`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- ne provoque pas l'appel du constructeur de copie de `C`
- provoque l'appel du destructeur `C`
- provoque une erreur de compilation
- provoque une erreur à l'exécution
- provoque l'appel du constructeur de copie de `C`

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge uniquement l'opérateur `<<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge les opérateurs `<` et `<<`
- surcharge uniquement l'opérateur `<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- à objets
- procédural
- fonctionnel

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre effectif transmis par référence
- un paramètre effectif transmis par valeur
- un paramètre formel transmis par valeur
- un paramètre formel transmis par référence

Question 3 En C++, la déclaration d'une classe :

- peut contenir 0 ou plusieurs constructeurs
- doit contenir au moins 1 constructeur
- ne doit contenir qu'1 seul constructeur
- doit toujours contenir 2 constructeurs

Question 4 En C++, on distingue des méthodes surchargées en fonction :

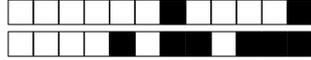
- du nombre ou du type de leurs paramètres
- uniquement du nombre de leurs paramètres
- de leurs noms
- de leurs types de retour

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this=this-c`
- `this-*c`
- `this-c`
- `this+(-c)`

Question 6 En C++, l'opérateur `new` permet de :

- définir un nouveau type alloué dynamiquement
- définir une nouvelle classe
- d'initialiser un nouveau type
- créer un objet alloué dynamiquement



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- rien du tout
- le destructeur la classe `C`
- le constructeur de copie de la classe `C`
- la surcharge de l'opérateur `=` de la classe `C`

Question 8 En C++, l'instruction `delete [] p;` :

- supprime les éléments du tableau `p` alloués dynamiquement
- déclare un tableau `p` de type `delete`
- supprime le premier élément du tableau `p`
- supprime la variable `p`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque une erreur à l'exécution
- ne provoque pas l'appel du constructeur de copie de `C`
- provoque une erreur de compilation
- provoque l'appel du constructeur de copie de `C`
- provoque l'appel du destructeur `C`

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge les opérateurs `<` et `<<`
- surcharge uniquement l'opérateur `<`
- surcharge uniquement l'opérateur `<<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- procédural
- à objets
- fonctionnel

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre effectif transmis par référence
- un paramètre formel transmis par référence
- un paramètre effectif transmis par valeur
- un paramètre formel transmis par valeur

Question 3 En C++, la déclaration d'une classe :

- doit contenir au moins 1 constructeur
- peut contenir 0 ou plusieurs constructeurs
- doit toujours contenir 2 constructeurs
- ne doit contenir qu'1 seul constructeur

Question 4 En C++, on distingue des méthodes surchargées en fonction :

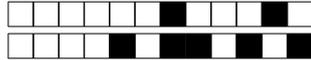
- de leurs noms
- du nombre ou du type de leurs paramètres
- de leurs types de retour
- uniquement du nombre de leurs paramètres

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this-c`
- `this-*c`
- `this=this-c`
- `this+(-c)`

Question 6 En C++, l'opérateur `new` permet de :

- d'initialiser un nouveau type
- créer un objet alloué dynamiquement
- définir un nouveau type alloué dynamiquement
- définir une nouvelle classe



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- rien du tout
- le destructeur la classe `C`
- la surcharge de l'opérateur `=` de la classe `C`
- le constructeur de copie de la classe `C`

Question 8 En C++, l'instruction `delete [] p;` :

- supprime la variable `p`
- supprime le premier élément du tableau `p`
- supprime les éléments du tableau `p` alloués dynamiquement
- déclare un tableau `p` de type `delete`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque l'appel du destructeur `C`
- ne provoque pas l'appel du constructeur de copie de `C`
- provoque une erreur de compilation
- provoque une erreur à l'exécution
- provoque l'appel du constructeur de copie de `C`

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge uniquement l'opérateur `<<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge uniquement l'opérateur `<`
- surcharge les opérateurs `<` et `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- à objets
- procédural
- fonctionnel

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre formel transmis par valeur
- un paramètre formel transmis par référence
- un paramètre effectif transmis par référence
- un paramètre effectif transmis par valeur

Question 3 En C++, la déclaration d'une classe :

- ne doit contenir qu'1 seul constructeur
- doit contenir au moins 1 constructeur
- doit toujours contenir 2 constructeurs
- peut contenir 0 ou plusieurs constructeurs

Question 4 En C++, on distingue des méthodes surchargées en fonction :

- de leurs noms
- uniquement du nombre de leurs paramètres
- du nombre ou du type de leurs paramètres
- de leurs types de retour

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this+(-c)`
- `this-*c`
- `this-c`
- `this=this-c`

Question 6 En C++, l'opérateur `new` permet de :

- créer un objet alloué dynamiquement
- définir un nouveau type alloué dynamiquement
- définir une nouvelle classe
- d'initialiser un nouveau type



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- le destructeur la classe `C`
- le constructeur de copie de la classe `C`
- rien du tout
- la surcharge de l'opérateur `=` de la classe `C`

Question 8 En C++, l'instruction `delete [] p;` :

- déclare un tableau `p` de type `delete`
- supprime la variable `p`
- supprime les éléments du tableau `p` alloués dynamiquement
- supprime le premier élément du tableau `p`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- ne provoque pas l'appel du constructeur de copie de `C`
- provoque l'appel du constructeur de copie de `C`
- provoque une erreur de compilation
- provoque une erreur à l'exécution
- provoque l'appel du destructeur `C`

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge uniquement l'opérateur `<<`
- surcharge uniquement l'opérateur `<`
- surcharge les opérateurs `<` et `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- procédural
- à objets
- fonctionnel

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre effectif transmis par référence
- un paramètre effectif transmis par valeur
- un paramètre formel transmis par valeur
- un paramètre formel transmis par référence

Question 3 En C++, la déclaration d'une classe :

- doit toujours contenir 2 constructeurs
- ne doit contenir qu'1 seul constructeur
- doit contenir au moins 1 constructeur
- peut contenir 0 ou plusieurs constructeurs

Question 4 En C++, on distingue des méthodes surchargées en fonction :

- du nombre ou du type de leurs paramètres
- de leurs noms
- de leurs types de retour
- uniquement du nombre de leurs paramètres

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this-c`
- `this+(-c)`
- `this=this-c`
- `this-*c`

Question 6 En C++, l'opérateur `new` permet de :

- d'initialiser un nouveau type
- définir un nouveau type alloué dynamiquement
- définir une nouvelle classe
- créer un objet alloué dynamiquement



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- le destructeur la classe `C`
- le constructeur de copie de la classe `C`
- la surcharge de l'opérateur `=` de la classe `C`
- rien du tout

Question 8 En C++, l'instruction `delete [] p;` :

- supprime la variable `p`
- supprime le premier élément du tableau `p`
- supprime les éléments du tableau `p` alloués dynamiquement
- déclare un tableau `p` de type `delete`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque une erreur de compilation
- provoque l'appel du constructeur de copie de `C`
- provoque une erreur à l'exécution
- ne provoque pas l'appel du constructeur de copie de `C`
- provoque l'appel du destructeur `C`

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge les opérateurs `<` et `<<`
- surcharge uniquement l'opérateur `<<`
- surcharge uniquement l'opérateur `<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- fonctionnel
- procédural
- à objets

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre formel transmis par valeur
- un paramètre formel transmis par référence
- un paramètre effectif transmis par valeur
- un paramètre effectif transmis par référence

Question 3 En C++, la déclaration d'une classe :

- ne doit contenir qu'1 seul constructeur
- peut contenir 0 ou plusieurs constructeurs
- doit contenir au moins 1 constructeur
- doit toujours contenir 2 constructeurs

Question 4 En C++, on distingue des méthodes surchargées en fonction :

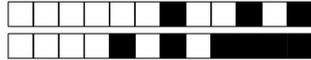
- du nombre ou du type de leurs paramètres
- de leurs noms
- de leurs types de retour
- uniquement du nombre de leurs paramètres

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this-*c`
- `this=this-c`
- `this+(-c)`
- `this-c`

Question 6 En C++, l'opérateur `new` permet de :

- définir une nouvelle classe
- définir un nouveau type alloué dynamiquement
- créer un objet alloué dynamiquement
- d'initialiser un nouveau type



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- rien du tout
- la surcharge de l'opérateur = de la classe `C`
- le constructeur de copie de la classe `C`
- le destructeur la classe `C`

Question 8 En C++, l'instruction `delete [] p;` :

- déclare un tableau `p` de type `delete`
- supprime la variable `p`
- supprime les éléments du tableau `p` alloués dynamiquement
- supprime le premier élément du tableau `p`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque l'appel du constructeur de copie de `C`
- provoque une erreur à l'exécution
- ne provoque pas l'appel du constructeur de copie de `C`
- provoque l'appel du destructeur `C`
- provoque une erreur de compilation

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge uniquement l'opérateur `<<`
- surcharge les opérateurs `<` et `<<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge uniquement l'opérateur `<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- fonctionnel
- procédural
- à objets

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre formel transmis par valeur
- un paramètre formel transmis par référence
- un paramètre effectif transmis par valeur
- un paramètre effectif transmis par référence

Question 3 En C++, la déclaration d'une classe :

- doit toujours contenir 2 constructeurs
- doit contenir au moins 1 constructeur
- peut contenir 0 ou plusieurs constructeurs
- ne doit contenir qu'1 seul constructeur

Question 4 En C++, on distingue des méthodes surchargées en fonction :

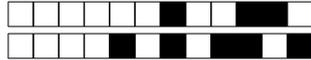
- de leurs types de retour
- du nombre ou du type de leurs paramètres
- uniquement du nombre de leurs paramètres
- de leurs noms

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this-c`
- `this=this-c`
- `this-*c`
- `this+(-c)`

Question 6 En C++, l'opérateur `new` permet de :

- définir une nouvelle classe
- créer un objet alloué dynamiquement
- définir un nouveau type alloué dynamiquement
- d'initialiser un nouveau type



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- le constructeur de copie de la classe `C`
- rien du tout
- le destructeur la classe `C`
- la surcharge de l'opérateur `=` de la classe `C`

Question 8 En C++, l'instruction `delete [] p;` :

- déclare un tableau `p` de type `delete`
- supprime les éléments du tableau `p` alloués dynamiquement
- supprime la variable `p`
- supprime le premier élément du tableau `p`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque l'appel du destructeur `C`
- provoque une erreur à l'exécution
- ne provoque pas l'appel du constructeur de copie de `C`
- provoque l'appel du constructeur de copie de `C`
- provoque une erreur de compilation

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge les opérateurs `<` et `<<`
- surcharge uniquement l'opérateur `<<`
- surcharge uniquement l'opérateur `<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- à objets
- procédural
- fonctionnel

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre effectif transmis par référence
- un paramètre effectif transmis par valeur
- un paramètre formel transmis par valeur
- un paramètre formel transmis par référence

Question 3 En C++, la déclaration d'une classe :

- ne doit contenir qu'1 seul constructeur
- doit contenir au moins 1 constructeur
- doit toujours contenir 2 constructeurs
- peut contenir 0 ou plusieurs constructeurs

Question 4 En C++, on distingue des méthodes surchargées en fonction :

- de leurs noms
- de leurs types de retour
- du nombre ou du type de leurs paramètres
- uniquement du nombre de leurs paramètres

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this=this-c`
- `this-c`
- `this-*c`
- `this+(-c)`

Question 6 En C++, l'opérateur `new` permet de :

- d'initialiser un nouveau type
- créer un objet alloué dynamiquement
- définir une nouvelle classe
- définir un nouveau type alloué dynamiquement



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- le constructeur de copie de la classe `C`
- rien du tout
- le destructeur la classe `C`
- la surcharge de l'opérateur `=` de la classe `C`

Question 8 En C++, l'instruction `delete [] p;` :

- supprime les éléments du tableau `p` alloués dynamiquement
- supprime la variable `p`
- déclare un tableau `p` de type `delete`
- supprime le premier élément du tableau `p`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque une erreur de compilation
- provoque l'appel du constructeur de copie de `C`
- provoque l'appel du destructeur `C`
- provoque une erreur à l'exécution
- ne provoque pas l'appel du constructeur de copie de `C`

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge uniquement l'opérateur `<`
- surcharge uniquement l'opérateur `<<`
- surcharge les opérateurs `<` et `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- à objets
- procédural
- fonctionnel

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre formel transmis par référence
- un paramètre effectif transmis par référence
- un paramètre effectif transmis par valeur
- un paramètre formel transmis par valeur

Question 3 En C++, la déclaration d'une classe :

- ne doit contenir qu'1 seul constructeur
- peut contenir 0 ou plusieurs constructeurs
- doit contenir au moins 1 constructeur
- doit toujours contenir 2 constructeurs

Question 4 En C++, on distingue des méthodes surchargées en fonction :

- de leurs noms
- du nombre ou du type de leurs paramètres
- de leurs types de retour
- uniquement du nombre de leurs paramètres

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this-*c`
- `this-c`
- `this=this-c`
- `this+(-c)`

Question 6 En C++, l'opérateur `new` permet de :

- définir une nouvelle classe
- définir un nouveau type alloué dynamiquement
- d'initialiser un nouveau type
- créer un objet alloué dynamiquement



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- la surcharge de l'opérateur = de la classe `C`
- le constructeur de copie de la classe `C`
- le destructeur la classe `C`
- rien du tout

Question 8 En C++, l'instruction `delete [] p;` :

- déclare un tableau `p` de type `delete`
- supprime le premier élément du tableau `p`
- supprime la variable `p`
- supprime les éléments du tableau `p` alloués dynamiquement

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

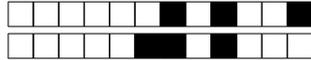
- ne provoque pas l'appel du constructeur de copie de `C`
- provoque l'appel du destructeur `C`
- provoque une erreur à l'exécution
- provoque une erreur de compilation
- provoque l'appel du constructeur de copie de `C`

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge les opérateurs `<` et `<<`
- surcharge uniquement l'opérateur `<<`
- surcharge uniquement l'opérateur `<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- à objets
- procédural
- fonctionnel

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre formel transmis par valeur
- un paramètre effectif transmis par valeur
- un paramètre effectif transmis par référence
- un paramètre formel transmis par référence

Question 3 En C++, la déclaration d'une classe :

- ne doit contenir qu'1 seul constructeur
- doit toujours contenir 2 constructeurs
- doit contenir au moins 1 constructeur
- peut contenir 0 ou plusieurs constructeurs

Question 4 En C++, on distingue des méthodes surchargées en fonction :

- de leurs types de retour
- de leurs noms
- du nombre ou du type de leurs paramètres
- uniquement du nombre de leurs paramètres

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this=this-c`
- `this-*c`
- `this-c`
- `this+(-c)`

Question 6 En C++, l'opérateur `new` permet de :

- créer un objet alloué dynamiquement
- d'initialiser un nouveau type
- définir un nouveau type alloué dynamiquement
- définir une nouvelle classe



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- la surcharge de l'opérateur = de la classe `C`
- le destructeur la classe `C`
- le constructeur de copie de la classe `C`
- rien du tout

Question 8 En C++, l'instruction `delete [] p;` :

- déclare un tableau `p` de type `delete`
- supprime le premier élément du tableau `p`
- supprime les éléments du tableau `p` alloués dynamiquement
- supprime la variable `p`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque une erreur à l'exécution
- ne provoque pas l'appel du constructeur de copie de `C`
- provoque une erreur de compilation
- provoque l'appel du constructeur de copie de `C`
- provoque l'appel du destructeur `C`

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge uniquement l'opérateur `<<`
- surcharge les opérateurs `<` et `<<`
- surcharge uniquement l'opérateur `<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- fonctionnel
- procédural
- à objets

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre formel transmis par valeur
- un paramètre formel transmis par référence
- un paramètre effectif transmis par valeur
- un paramètre effectif transmis par référence

Question 3 En C++, la déclaration d'une classe :

- doit toujours contenir 2 constructeurs
- ne doit contenir qu'1 seul constructeur
- doit contenir au moins 1 constructeur
- peut contenir 0 ou plusieurs constructeurs

Question 4 En C++, on distingue des méthodes surchargées en fonction :

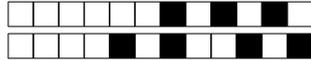
- de leurs noms
- de leurs types de retour
- du nombre ou du type de leurs paramètres
- uniquement du nombre de leurs paramètres

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this-c`
- `this+(-c)`
- `this-*c`
- `this=this-c`

Question 6 En C++, l'opérateur `new` permet de :

- définir un nouveau type alloué dynamiquement
- créer un objet alloué dynamiquement
- définir une nouvelle classe
- d'initialiser un nouveau type



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- la surcharge de l'opérateur `=` de la classe `C`
- rien du tout
- le constructeur de copie de la classe `C`
- le destructeur la classe `C`

Question 8 En C++, l'instruction `delete [] p;` :

- supprime le premier élément du tableau `p`
- déclare un tableau `p` de type `delete`
- supprime la variable `p`
- supprime les éléments du tableau `p` alloués dynamiquement

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque l'appel du constructeur de copie de `C`
- provoque une erreur de compilation
- provoque une erreur à l'exécution
- provoque l'appel du destructeur `C`
- ne provoque pas l'appel du constructeur de copie de `C`

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge uniquement l'opérateur `<<`
- surcharge uniquement l'opérateur `<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge les opérateurs `<` et `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- à objets
- fonctionnel
- procédural

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre formel transmis par référence
- un paramètre formel transmis par valeur
- un paramètre effectif transmis par référence
- un paramètre effectif transmis par valeur

Question 3 En C++, la déclaration d'une classe :

- doit contenir au moins 1 constructeur
- peut contenir 0 ou plusieurs constructeurs
- ne doit contenir qu'1 seul constructeur
- doit toujours contenir 2 constructeurs

Question 4 En C++, on distingue des méthodes surchargées en fonction :

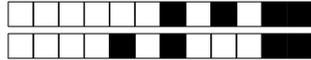
- de leurs noms
- du nombre ou du type de leurs paramètres
- de leurs types de retour
- uniquement du nombre de leurs paramètres

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this=this-c`
- `this-c`
- `this+(-c)`
- `this-*c`

Question 6 En C++, l'opérateur `new` permet de :

- d'initialiser un nouveau type
- définir un nouveau type alloué dynamiquement
- créer un objet alloué dynamiquement
- définir une nouvelle classe



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- le constructeur de copie de la classe `C`
- la surcharge de l'opérateur `=` de la classe `C`
- le destructeur la classe `C`
- rien du tout

Question 8 En C++, l'instruction `delete [] p;` :

- supprime les éléments du tableau `p` alloués dynamiquement
- déclare un tableau `p` de type `delete`
- supprime la variable `p`
- supprime le premier élément du tableau `p`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque une erreur à l'exécution
- provoque une erreur de compilation
- ne provoque pas l'appel du constructeur de copie de `C`
- provoque l'appel du destructeur `C`
- provoque l'appel du constructeur de copie de `C`

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge uniquement l'opérateur `<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge les opérateurs `<` et `<<`
- surcharge uniquement l'opérateur `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- à objets
- procédural
- fonctionnel

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre formel transmis par valeur
- un paramètre effectif transmis par référence
- un paramètre effectif transmis par valeur
- un paramètre formel transmis par référence

Question 3 En C++, la déclaration d'une classe :

- doit contenir au moins 1 constructeur
- ne doit contenir qu'1 seul constructeur
- peut contenir 0 ou plusieurs constructeurs
- doit toujours contenir 2 constructeurs

Question 4 En C++, on distingue des méthodes surchargées en fonction :

- de leurs noms
- uniquement du nombre de leurs paramètres
- du nombre ou du type de leurs paramètres
- de leurs types de retour

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this+(-c)`
- `this-*c`
- `this=this-c`
- `this-c`

Question 6 En C++, l'opérateur `new` permet de :

- créer un objet alloué dynamiquement
- définir une nouvelle classe
- définir un nouveau type alloué dynamiquement
- d'initialiser un nouveau type



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- rien du tout
- le constructeur de copie de la classe `C`
- la surcharge de l'opérateur `=` de la classe `C`
- le destructeur la classe `C`

Question 8 En C++, l'instruction `delete [] p;` :

- déclare un tableau `p` de type `delete`
- supprime la variable `p`
- supprime les éléments du tableau `p` alloués dynamiquement
- supprime le premier élément du tableau `p`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque l'appel du destructeur `C`
- provoque l'appel du constructeur de copie de `C`
- provoque une erreur à l'exécution
- ne provoque pas l'appel du constructeur de copie de `C`
- provoque une erreur de compilation

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge les opérateurs `<` et `<<`
- surcharge uniquement l'opérateur `<`
- surcharge uniquement l'opérateur `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- fonctionnel
- à objets
- procédural

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre effectif transmis par référence
- un paramètre formel transmis par valeur
- un paramètre effectif transmis par valeur
- un paramètre formel transmis par référence

Question 3 En C++, la déclaration d'une classe :

- ne doit contenir qu'1 seul constructeur
- peut contenir 0 ou plusieurs constructeurs
- doit toujours contenir 2 constructeurs
- doit contenir au moins 1 constructeur

Question 4 En C++, on distingue des méthodes surchargées en fonction :

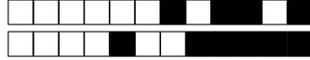
- uniquement du nombre de leurs paramètres
- de leurs types de retour
- du nombre ou du type de leurs paramètres
- de leurs noms

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this-c`
- `this+(-c)`
- `this=this-c`
- `this-*c`

Question 6 En C++, l'opérateur `new` permet de :

- créer un objet alloué dynamiquement
- définir un nouveau type alloué dynamiquement
- définir une nouvelle classe
- d'initialiser un nouveau type



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- rien du tout
- la surcharge de l'opérateur `=` de la classe `C`
- le constructeur de copie de la classe `C`
- le destructeur la classe `C`

Question 8 En C++, l'instruction `delete [] p;` :

- supprime le premier élément du tableau `p`
- supprime la variable `p`
- supprime les éléments du tableau `p` alloués dynamiquement
- déclare un tableau `p` de type `delete`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

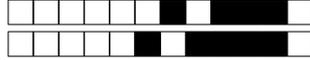
- provoque l'appel du destructeur `C`
- ne provoque pas l'appel du constructeur de copie de `C`
- provoque une erreur de compilation
- provoque l'appel du constructeur de copie de `C`
- provoque une erreur à l'exécution

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge uniquement l'opérateur `<`
- surcharge uniquement l'opérateur `<<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge les opérateurs `<` et `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- procédural
- fonctionnel
- à objets

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre effectif transmis par référence
- un paramètre effectif transmis par valeur
- un paramètre formel transmis par valeur
- un paramètre formel transmis par référence

Question 3 En C++, la déclaration d'une classe :

- doit contenir au moins 1 constructeur
- doit toujours contenir 2 constructeurs
- peut contenir 0 ou plusieurs constructeurs
- ne doit contenir qu'1 seul constructeur

Question 4 En C++, on distingue des méthodes surchargées en fonction :

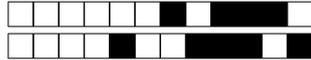
- de leurs types de retour
- uniquement du nombre de leurs paramètres
- du nombre ou du type de leurs paramètres
- de leurs noms

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this=this-c`
- `this-c`
- `this-*c`
- `this+(-c)`

Question 6 En C++, l'opérateur `new` permet de :

- créer un objet alloué dynamiquement
- définir une nouvelle classe
- d'initialiser un nouveau type
- définir un nouveau type alloué dynamiquement



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- le constructeur de copie de la classe `C`
- la surcharge de l'opérateur `=` de la classe `C`
- le destructeur la classe `C`
- rien du tout

Question 8 En C++, l'instruction `delete [] p;` :

- déclare un tableau `p` de type `delete`
- supprime le premier élément du tableau `p`
- supprime la variable `p`
- supprime les éléments du tableau `p` alloués dynamiquement

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

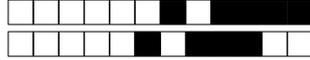
- provoque l'appel du destructeur `C`
- provoque une erreur à l'exécution
- ne provoque pas l'appel du constructeur de copie de `C`
- provoque une erreur de compilation
- provoque l'appel du constructeur de copie de `C`

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge uniquement l'opérateur `<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge uniquement l'opérateur `<<`
- surcharge les opérateurs `<` et `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- procédural
- à objets
- fonctionnel

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre formel transmis par référence
- un paramètre formel transmis par valeur
- un paramètre effectif transmis par référence
- un paramètre effectif transmis par valeur

Question 3 En C++, la déclaration d'une classe :

- doit toujours contenir 2 constructeurs
- peut contenir 0 ou plusieurs constructeurs
- ne doit contenir qu'1 seul constructeur
- doit contenir au moins 1 constructeur

Question 4 En C++, on distingue des méthodes surchargées en fonction :

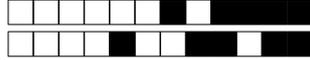
- uniquement du nombre de leurs paramètres
- de leurs noms
- du nombre ou du type de leurs paramètres
- de leurs types de retour

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this-*c`
- `this=this-c`
- `this+(-c)`
- `this-c`

Question 6 En C++, l'opérateur `new` permet de :

- définir une nouvelle classe
- définir un nouveau type alloué dynamiquement
- d'initialiser un nouveau type
- créer un objet alloué dynamiquement



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- rien du tout
- la surcharge de l'opérateur `=` de la classe `C`
- le destructeur la classe `C`
- le constructeur de copie de la classe `C`

Question 8 En C++, l'instruction `delete [] p;` :

- supprime les éléments du tableau `p` alloués dynamiquement
- supprime la variable `p`
- supprime le premier élément du tableau `p`
- déclare un tableau `p` de type `delete`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque l'appel du constructeur de copie de `C`
- provoque une erreur à l'exécution
- provoque une erreur de compilation
- provoque l'appel du destructeur `C`
- ne provoque pas l'appel du constructeur de copie de `C`

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge uniquement l'opérateur `<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge uniquement l'opérateur `<<`
- surcharge les opérateurs `<` et `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- procédural
- à objets
- fonctionnel

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre formel transmis par référence
- un paramètre effectif transmis par valeur
- un paramètre effectif transmis par référence
- un paramètre formel transmis par valeur

Question 3 En C++, la déclaration d'une classe :

- doit contenir au moins 1 constructeur
- peut contenir 0 ou plusieurs constructeurs
- ne doit contenir qu'1 seul constructeur
- doit toujours contenir 2 constructeurs

Question 4 En C++, on distingue des méthodes surchargées en fonction :

- uniquement du nombre de leurs paramètres
- du nombre ou du type de leurs paramètres
- de leurs noms
- de leurs types de retour

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this-c`
- `this-*c`
- `this+(-c)`
- `this=this-c`

Question 6 En C++, l'opérateur `new` permet de :

- d'initialiser un nouveau type
- définir une nouvelle classe
- définir un nouveau type alloué dynamiquement
- créer un objet alloué dynamiquement



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- le destructeur la classe `C`
- le constructeur de copie de la classe `C`
- rien du tout
- la surcharge de l'opérateur `=` de la classe `C`

Question 8 En C++, l'instruction `delete [] p;` :

- supprime la variable `p`
- supprime le premier élément du tableau `p`
- déclare un tableau `p` de type `delete`
- supprime les éléments du tableau `p` alloués dynamiquement

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- ne provoque pas l'appel du constructeur de copie de `C`
- provoque une erreur de compilation
- provoque une erreur à l'exécution
- provoque l'appel du constructeur de copie de `C`
- provoque l'appel du destructeur `C`

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge uniquement l'opérateur `<<`
- surcharge les opérateurs `<` et `<<`
- surcharge uniquement l'opérateur `<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- à objets
- procédural
- fonctionnel

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre formel transmis par valeur
- un paramètre effectif transmis par référence
- un paramètre effectif transmis par valeur
- un paramètre formel transmis par référence

Question 3 En C++, la déclaration d'une classe :

- ne doit contenir qu'1 seul constructeur
- peut contenir 0 ou plusieurs constructeurs
- doit toujours contenir 2 constructeurs
- doit contenir au moins 1 constructeur

Question 4 En C++, on distingue des méthodes surchargées en fonction :

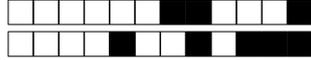
- de leurs noms
- uniquement du nombre de leurs paramètres
- du nombre ou du type de leurs paramètres
- de leurs types de retour

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this-*c`
- `this-c`
- `this+(-c)`
- `this=this-c`

Question 6 En C++, l'opérateur `new` permet de :

- définir un nouveau type alloué dynamiquement
- d'initialiser un nouveau type
- créer un objet alloué dynamiquement
- définir une nouvelle classe



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- rien du tout
- la surcharge de l'opérateur = de la classe `C`
- le destructeur la classe `C`
- le constructeur de copie de la classe `C`

Question 8 En C++, l'instruction `delete [] p;` :

- supprime le premier élément du tableau `p`
- déclare un tableau `p` de type `delete`
- supprime les éléments du tableau `p` alloués dynamiquement
- supprime la variable `p`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque une erreur de compilation
- ne provoque pas l'appel du constructeur de copie de `C`
- provoque l'appel du destructeur `C`
- provoque l'appel du constructeur de copie de `C`
- provoque une erreur à l'exécution

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- surcharge uniquement l'opérateur `<`
- surcharge les opérateurs `<` et `<<`
- surcharge uniquement l'opérateur `<<`
- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`



DS du 24/11/2016

Durée 1h

Nom :

Prénom :

ID :

Aucun document n'est autorisé. L'usage du téléphone mobile est interdit.

Les questions ne comportent qu'une seule bonne réponse. On retire 1 point par réponse fausse cochée.

Question 1 C++ est un langage :

- procédural
- à objets
- fonctionnel

Question 2 En C++, dans la en-tête de la fonction `void f(int &x)`, `x` est :

- un paramètre effectif transmis par référence
- un paramètre effectif transmis par valeur
- un paramètre formel transmis par valeur
- un paramètre formel transmis par référence

Question 3 En C++, la déclaration d'une classe :

- peut contenir 0 ou plusieurs constructeurs
- doit toujours contenir 2 constructeurs
- ne doit contenir qu'1 seul constructeur
- doit contenir au moins 1 constructeur

Question 4 En C++, on distingue des méthodes surchargées en fonction :

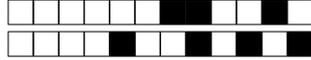
- de leurs noms
- du nombre ou du type de leurs paramètres
- uniquement du nombre de leurs paramètres
- de leurs types de retour

Question 5 En C++, le prototype `C C::operator-(C &c) const` renvoie :

- `this-*c`
- `this-c`
- `this+(-c)`
- `this=this-c`

Question 6 En C++, l'opérateur `new` permet de :

- définir un nouveau type alloué dynamiquement
- d'initialiser un nouveau type
- définir une nouvelle classe
- créer un objet alloué dynamiquement



Question 7 En C++, dans l'instruction : `C c1; C c2=c1;`, la déclaration de `c2` fait intervenir :

- rien du tout
- la surcharge de l'opérateur `=` de la classe `C`
- le destructeur la classe `C`
- le constructeur de copie de la classe `C`

Question 8 En C++, l'instruction `delete [] p;` :

- supprime les éléments du tableau `p` alloués dynamiquement
- supprime la variable `p`
- supprime le premier élément du tableau `p`
- déclare un tableau `p` de type `delete`

Question 9 En C++, les instructions `C x = C(); f(x);` avec les déclarations :

```
class C {};  
void f(C &c) {}
```

- provoque l'appel du constructeur de copie de `C`
- provoque l'appel du destructeur `C`
- provoque une erreur à l'exécution
- ne provoque pas l'appel du constructeur de copie de `C`
- provoque une erreur de compilation

Question 10 Avec la méthode vue en cours :

```
template <typename T>  
const T& minimum(const T& x, const T& y)
```

l'énoncé `cout << minimum(rectangle(2,4),rectangle(3,5));` est valide si la classe `rectangle` :

- ne surcharge *ni* l'opérateur `<`, *ni* l'opérateur `<<`
- surcharge uniquement l'opérateur `<`
- surcharge uniquement l'opérateur `<<`
- surcharge les opérateurs `<` et `<<`