

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

CORRECTION

**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 `<<`