



Examen en « Program Orientée Objet en C++ »

Durée 1h30

- Si on n'utilise pas l'expression `using namespace std`, que faut-il ajouté à ce code :
`cout << "Hello" << endl`
- Pour tous les noms de variables suivants, choisissez ceux qui sont acceptés par le compilateur :

2eme_element	main1	include	DELETE	var.2
--------------	-------	---------	--------	-------
- Quelle est la différence entre les deux expressions :
`int x = &a ;` et `int &x = a ;`
- Citer les 3 modes de passage de paramètre dans une fonction.
- Donner l'opération qui permet de détruire l'allocation dynamique « `ptr=new int[100]` ».
- Avec quel mode de passage de paramètre, un tableau est-il passé à une fonction ?
- Citer les quatre principes fondamentaux du concept de la programmation orienté objet.
- Donner les caractéristiques d'un constructeur de classe en C++.
- Quels sont les types d'héritage possible en C++ ?
- Qu'est-ce qu'une classe abstraite ?
- Quelle est la ou les bonnes expressions qui permettent de créer un objet de type Point.
`Point (a);` `Point = a ;` `Point *a ;` `Point a ;`
- Qu'affiche ce programme ?

```
int a(16); int b(a + 7);
a /= 2;
cout << "a" << ", " << b << endl;
```
- Qu'affiche ce programme ?

```
int a(1);
for (int i(0); i < 4; i += 1)
{
    cout << i << " " << a << " ";
    if (i % 2 == 1)
        a += i;
}
```
- Qu'affiche ce programme ?

```
Int add(int a, int &b)
{
    return (a+b++);
}
void main()
{
    int x=2, y=3; cout << x << "+" << y << "=" << add(x,y) ;
}
```
- Qu'affiche ce programme ?

```
void main(){
    int i=2 ;
    switch(i) {
        case 1 : cout << "1" ;
        case 2 : cout << "2" ;
        case 3 : cout << "3" ;
        case 4 : cout << "4" ;
    }
```
- En C++, une fonction ne peut pas retourner un pointeur (Vrai ou faux et justifier).
- Si une fonction ne prend aucun argument, sans type de retour est `void` (Vrai ou faux et justifier).
- Une fonction amie peut être déclarée `public` ou `private` (Vrai ou faux et justifier).
- Utiliser un `break` dans un `switch` est optionnel (Vrai ou faux).
- Un constructeur peut être virtuel (Vrai ou faux).