



# Devoir Surveillé de C++ n°1

Enseignant : Nils Beaussé

---

Calculatrice **interdite** | Durée 2h : 8h – 10h | **Bon courage à tous !**

---



## Partie 1 – Questions de cours : (15 pts)

- (0.5 pt)** Expliquez la différence entre langage interprété et non interprété ainsi qu'entre langage bas-niveau et haut-niveau.
- (1 pt)** Résumez avec vos mots les principales différences entre le C et le C++
- (1 pt)** Parmi les assertions suivantes, notez lesquelles sont vraies et lesquelles sont fausses :
  - Un fichier en C doit être d'extension « .c ».
  - Un fichier en C++ doit être d'extension « .cpp ».
  - Un fichier « .o » est un fichier qui contient des bouts de code sous forme de texte.
  - L'étape d'édition des liens (« linkage ») est une étape qui lie les fichiers « .cpp » entre eux.
- (0,5 pt)** Décrivez les différentes étapes de la compilation d'un programme à l'aide de GCC (ou G++)
- (0,5 pt)** Donner la ligne de commande permettant de compiler un programme composé de trois fichiers avec GCC (ou G++).
  - En une seule commande.
  - En plusieurs commandes.
- (0.5 pt)** Les includes en C++ qui sont issus du C s'écrivent d'une autre manière, traduire les includes ci-dessous :
  - stdio.h
  - stdlib.h
  - string.h
- (0,5 pt)** Décrivez l'usage du mot clé « auto », en quoi est-il dangereux ?

8. (0.5 pt) Qu'affiche le code suivant :

```
#include <iostream>

int main ()
{
    bool b = true;
    int i;
    std::cout << b << std::endl;
    std::cout << std::boolalpha << b << '\n';
    std::cout << std::noboolalpha << b << '\n';
    i = b;
    std::cout << i << std::endl;
    i = -4;
    b = i;
    std::cout << b << std::endl;
    i = 0;
    b = i;
    std::cout << b << std::endl;

    return 0;
}
```

9. (0.5 pt) Une erreur s'est glissée dans ce code, identifiez là, expliquez en quoi elle consiste et ce qu'il faudrait faire pour la corriger.

```
#include <iostream>
using namespace std

namespace f2
{
    int inc(int n) {
        return n+1;
    }
}

int main ()
{
    cout << inc(5) << endl;
}
```

10. (1 pt) Rappelez comment on utilise malloc et new pour allouer un tableau de 20 char.
11. (1 pt) Expliquez avec vos mots : pourquoi faut-il utiliser new plutôt que malloc en C++ ?



12. (2 pt) Expliquez l'usage de chacun des bouts de code ci-dessous, ainsi que le vocabulaire qui y est associé. Détaillez ce qui se passe à bas niveau avec chacun de ces codes et à chaque étape :

a) `int* a = 0;`

b) `int c;`  
`int* a = &c;`  
`*a = 2;`  
`c = 4;`  
`a = 0;`

c) `int b;`  
`int& e = b;`  
`e = 3;`  
`b = 5;`

d) `int a;`  
`int &b = a;`  
`int* c = &b;`  
  
`*c = 2;`  
`b = 4;`  
`a = 5;`  
`c = 6;`

13. (0,5 pt) Soit une classe « personnage », comment écririez-vous le nom du constructeur par défaut de cette classe ?

14. (0,5) Si je crée un constructeur avec des arguments en n'écrivant pas le constructeur par défaut, quel est le risque ?

15. (1 pt) Si je ne crée pas de constructeur de copie, dans quel cas est-ce un gros risque ?

1. (3.5 pts) Le code suivant comporte plusieurs erreurs, quelles sont-elles et pourquoi ?

```
class Notes{
    int* tab;
    int taille = 0;
    double devoir;
    double exam;
    double noteS3(){return 0.5*devoir+0.5*exam;}

    Notes(){
        tab = new int[taille];
        devoir = 0.0;
        exam = 0.0;
    }

    Notes(int a){
        taille = a;
        tab = new int[taille];
        devoir = 0.0;
        exam = 0.0;
    }

    Notes(int a, double b = 0.0, double exam = 0.0) : taille(a){
        tab = new int[taille];
        devoir = b;
        exam = exam;
    }

    ~Notes(){
        delete tab;
    }
};

int main(){
    Notes notes;
    Notes notes2(4);

    notes.exam = 14;
    notes.devoir = 4;

    notes2.tab[0] = 1;

    notes.noteS3();
    notes2.noteS3();

    return 0;
}
```



## Partie 2 – Exercices de compositions : (5 pts)

### Exercice 1 : (2.5 pts)

Proposez un ensemble de fonctions en C++ capable d'effectuer l'addition de 1 à 4 entiers, de 1 à 4 flottants et de 1 à 4 doubles. Ces fonctions doivent être capables de renvoyer le résultat. Le code devra être le plus court et le plus simple possible.

### Exercice 2 : (2.5 pts)

Créer une classe personnage comprenant un tableau de char pour son nom, un int pour sa vie et un deuxième int pour sa puissance.

La classe devra être séparé en deux partie : une pour le fichier d'en tête et une pour le fichier de code. Vous donnerez un exemple d'usage de la classe dans un main.

- La classe devra contenir des accesseurs et mutateurs.
- Des initialisations par défauts
- Une fonction d'initialisation
- Une fonction permettant de taper sur un autre personnage.
- Une fonction ultime\_setter pour les réunir tous et dans les ténèbres les lier. Elle devra pouvoir prendre tout les paramètre de la classe pour les modifier en un coup.

BONUS : Créez les constructeur et destructeurs de la manière la plus pertinente par rapport à ce que vous avez vu en cours.