



Programmation Orientée Objet 2024-2025

DST2

- vérifiez que vous avez six pages numérotées de 1 à 6 portant toutes le même numéro de sujet (en haut à droite où S : numéro de sujet, N : numéro de page) - les notes de cours uniquement personnelles sont autorisées - les appareils électroniques susceptibles de communiquer sont interdits - indiquez votre nom et le numéro du sujet sur tout rendu de document annexe - indiquez le nombre de feuilles que vous rendez - si vous répondez sur un document annexe alors marquez-le sur le sujet à l'endroit prévu pour la réponse - indiquez le numéro de la question au dessus de votre réponse en cas d'ambiguïté - rendez les six pages du sujet avec l'entête complété dans tous les cas -

0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9

← Codez votre numéro d'étudiant ci-contre en noircissant un chiffre par colonne. (si vous vous trompez écrivez votre numéro en dessous)

nom et prénom :

vous groupe :

nombre de feuilles rendues :

Indications : Par défaut les extraits de code sources du sujet sont écrits en langage C++. Par défaut aucune justification n'est demandée.

Question 1 On appelle MagmaR un ensemble à 3 éléments : $\{0, 1, 2\}$ muni d'une opération (*) bizarre. Voici la table de l'opération du MagmaR

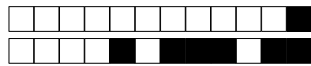
*	0	1	2
0	1	0	2
1	1	0	2
2	2	1	1

On souhaite créer une classe MagmaR qui permet de manipuler des éléments de la classe. Voici un exemple d'utilisation :

```
MagmaR a{0};
MagmaR b{1};
// MagmaR b{3}; // doit signaler une erreur (methode de votre choix) car il n'y a pas de valeur 3

MagmaR c=a*b;
// doit renvoyer une instance magmaR initialisee avec 0
// (car 0*1 donne 0 selon la table)
MagmaR c=b*a;
b==a; // doit renvoyer faux (ou 0)
a!=b; // doit renvoyer vrai (ou 1)
```

Rédiger la classe sur une feuille annexe.

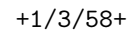


Question 2 Voici un code C++ qui produit une erreur lors de sa compilation. Indiquez à quelle ligne se trouve l'erreur et indiquez la cause de cette erreur.

```
class A{
    int a;
public:
    static void f(){
        g(); }
    void g(){}
};
int main() {
    A a;
    a.g();
    a.f();
    A::f();
    return 0;
}
```

☐N ☐QN ☐M ☐AB ☐B ☐TB ☐P

.....
.....
.....
.....



```
#include<iostream>
struct A{
    void f(){std::cout<< "fA" << std::endl;}
    virtual void g(){std::cout<< "gA" << std::endl;}
};
struct B: public A{
    virtual void f(){std::cout<< "fB" << std::endl;}
    void g(){std::cout<< "gB" << std::endl;}
};
int main(){
    B ob;
    A& ra=ob;
    A* pa=&ob;
    pa->f();
    pa->g();
    ra.f();
    ra.g();
    ob.f();
    ob.g();
    return 0;
}
```

☐N ☐QN ☐M ☐AB ☐B ☐TB ☐P

[illegible]

Question 4 Voici un code qui compile et s'exécute correctement. Indiquer ce qu'il affiche.

```
#include<iostream>

class Ma {
    int a;
public:
    Ma(int b): a{b}{
        std::cout << "construction MA" << std::endl;
    }
    void f(){
        std::cout << a << std::endl;
    }
};

int main()
{
    Ma ob1{3};
    Ma ob2{4};
    ob1.f();
    ob2.f();

    return 0;
}
```

☐N ☐QN ☐M ☐AB ☐B ☐TB ☐P

[illegible]

Question 5 Voici un code qui compile et s'exécute correctement.

1. Pour chaque ligne de la fonction main indiquer ce qu'elle affiche. (L1,L2,...)
2. Si on décommente la ligne commentée (L8) le code est-t-il encore compilable ? Si oui qu'affiche la ligne décommenté, si non pourquoi ?

```
#include<iostream>
struct A{
    int a;
    A(int p):a{p}{std::cout << "A" << a <<" ";}
    ~A(){std::cout << "~A" << a <<" ";}  };
struct C{
    int c;
    C():c{1}{std::cout << "C" << 1 <<" ";}
    C(int p):c{p}{std::cout << "C" << c <<" ";}
    ~C(){std::cout << "~C" << c <<" ";}  };
struct B : public A, public C{
    A a{16};
    C c;
    int b;
    B(int p):A{p+2},C{p+4},c{p+8},b{p}{std::cout << "B" << b <<" ";}
    ~B(){std::cout << "~B" << b <<" ";}  };
int main(){
    A oba{1};    //L1
    C* obc;      //L2
    C obc2{5};   //L3
    A oba2=oba;  //L4
    {            //L5
        C &obc3=obc2; //L6
        B obb{4};     //L7
        // A oba2;     //L8
        C* obc4=new C; //L9
    }              //L10
    std::cout << "fin "; //L11
    return 0;       //L12
}
```

☐ N ☐ QN ☐ M ☐ AB ☐ B ☐ TB ☐ P

[illegible]



Question 6 Écrire les trois classes suivantes sans utiliser d'outils provenant de bibliothèques.

1. Une classe `Tableau2dInt` dont les objets sont des tableaux à deux dimensions contenant des valeurs de type `int`, on peut comparer deux tableaux avec les opérateurs `==` et `!=` (attentions aux dimensions). On pourra modifier la valeur d'une case ou récupérer la valeur d'une case (attentions aux dimensions).
2. Une classe `Tableau2dDouble` aux mêmes capacités que la classe précédente mais avec des cases de type `double`
3. Une classe `MatriceInt` (des matrices d'entiers dont on peut faire la somme) aux mêmes capacités que `Tableau2dInt` avec en plus la possibilité de faire la somme de deux Tableaux si leurs dimensions sont les mêmes.

Exemple d'utilisation :

```
#include<iostream>
/* ici définition des classes
    Tableau2dInt
    Tableau2dDouble
    MatriceInt */
int main(){
    Tableau2dInt a{3,2,0}; // tableau de int à 2 dimensions, 3 lignes, 2 colonnes, rempli par des 0
    Tableau2dDouble b{4,5,1.3}; // tableau de double, 4 lignes, 5 colonnes, rempli par 1.3

    b.set(3,4,3.5); // modifie la case d'indice (3,4) (4ieme ligne,5ieme colonne)
    int r=a.get(1,1); // renvoie la case d'indice (1,1)
    a!=a; // renvoie faux (ou 0)

    MatriceInt c{4,2,4};
    MatriceInt d{4,2,7};
    MatriceInt e=c+d;
    MatriceInt f{4,2,11};
    e==f; // renvoie vrai (ou 1)
    return 0; }
```

Indications :

- Vous pouvez construire des classes et fonctions supplémentaires
- Vous pouvez utiliser les outils d'héritage ou des patrons de classe.
- Vous devez penser à la gestion de la mémoire et implanter : destructeur, constructeur de copie, affectation, déplacement, constructeur par déplacement si cela a un sens.
- Pensez aux objets constants.
- Pensez aux natures des membres : privé ou public (ou protected).

☐ N ☐ QN ☐ M ☐ AB ☐ B ☐ TB ☐ P