

## Examen M1 / C++

*La propreté de la programmation sera pris en compte dans l'évaluation.*

**Exercice 1** Quels sont les paradigmes supportés par C++?

**Exercice 2** Donnez le code de la fonction `triple` transformant un nombre  $n$  en son triple  $3 \times n$ , par exemple on déclare `int x=4;` puis après l'appel on a  $x = 12$ , dans les cas suivants:

- 1) L'appel avait été effectué par l'instruction `triple(&x);`
- 2) L'appel avait été effectué par l'instruction `triple(x);`
- 3) L'appel avait été effectué par l'instruction `x=triple(x);`

**Exercice 3** 1) Implémenter une classe `Rationnels` modélisant l'ensemble  $\mathbb{Q} = \{p/q : p \in \mathbb{Z}, q \in \mathbb{N} \setminus \{0\}\}$  ayant comme attributs privés `int p,q;` et ayant deux opérateurs membres `+=` et `*=` modélisant l'addition et la multiplication dans  $\mathbb{Q}$ . Par exemple:

```
Rationnels a(2,3),b(1,5),c(3,1),d(12,15);  
a*=b; // a=2/15
```

2) Implémenter des opérateurs binaires `+` et `*` modélisant l'addition et la multiplication dans  $\mathbb{Q}$  mais n'ayant pas accès aux attributs privés de `Rationnels`. Les opérateurs binaires devront supporter le chaînage, par exemple on doit pouvoir faire:

```
a= a+b+c; // a=28/15
```

3) Faire une fonction membre `reduct` qui réduit la fraction. Par exemple

```
d.reduct(); // d=4/5
```

**Exercice 4** Créer une classe `vect` simulant un vecteur de dimension variable. On mettra en attribut privé `int dim` la dimension du vecteur et `double * adr` l'adresse du vecteur (dont les composantes seront de type `double`). Faire les méthodes suivantes pour gérer la partie dynamique et libérer l'espace mémoire : un constructeur prenant la dimension en paramètre, un destructeur, un constructeur de copie et l'opérateur `=` surdéfini.