

# Programmation Fonctionnelle (2024-2025)

## CC 1

### Exercice 1 (/5)

Soit le code suivant :

```
public class Personne {  
    Ê public Adresse adresse(CountryCode cc) { ... }  
    Ê public static CountryCode code(Country c) { ... }  
    Ê public static Function<CountryCode, List<String>> cities = ... ;  
}
```

! Donnez le type (notation du cours) pour `adresse`, `code` et `ci ti es`.

Soit le code suivant :

```
object Calculatrice {  
    Ê def tradCurrency(source: Currency, target: Currency, amount: Double): Double =...  
    Ê def tradCurrency2(source: Currency, target: Currency)(amount: Double): Double =...  
}
```

! Donnez le type (notation du cours) pour `tradCurrency` et `tradCurrency2`.

### Exercice 2 (/5)

Les mĳthodes suivantes sont elles pures ou non (dans ce cas, dites pourquoi) ?

```
static double salaire1(double bonus) { return 2000 + bonus; }  
static double salaire2(double bonus) { return 2000 + 100 * Math.random(); }  
public int recruter(Personne salariŽ) { this.salariŽs.add(salariŽ); return salariŽs  
.size(); }  
public static List<Personne> recruter2(List<Personne> salariŽs, Personne salariŽ) {  
return new ArrayList<>(salariŽs).add(salariŽ); }  
public static int rabais(List<Book> books) { return books.stream().filter(b -> b.  
price() > 100).count(); }
```

### Exercice 3 (/5)

! Refactorer.

```
public class Promotion {
```

```

Ê private List<Etudiants> žtudiants;
Ê public Promotion() { this.žtudiants = new ArrayList<>(); }
Ê public void inscrire(Etudiant žtudiant) { this.žtudiants.add(žtudiant); }
Ê public int nombreEtudiants() { return this.žtudiants.size(); }
}

```

## Exercice 4 (/5)

Soit le code suivant :

```

case class Etudiant(nom: String, prenom: String, age: Int, nbUE: Int, note: Double);
val žtudiants = List(...)

```

! Ecrivez en Scala les mžthodes pour :

- ¥ obtenir la liste des notes des žtudiants.
- ¥ obtenir la liste des žtudiants trižs par note (croissante).
- ¥ obtenir la liste des žtudiants ayant la moyenne (note supžrieure ou žgale ^ 10.0)
- ¥ obtenir le nombre džtudiants ayant une note supžrieure ou žgale ^ 12.5.
- ¥ obtenir le nombre total dŃUE prises par lŃensemble des žtudiants.

Vous pourrez avoir ^ džfinir des mžthodes / fonctions auxiliaires si nžcessaire.