

Corrigé de l'examen 1 : Master 1 ESE+Inst
Programmation Orientée Objet

Questions de cours 5 pts

- 1- En programmation structurée, un programme est formé de la réunion de différentes procédures et de différentes structures de données, D'autre part, les procédures utilisent un certain nombre de structures de contrôle bien définies.

La programmation orientée objet (en abrégé P.O.O) est fondée sur le concept **d'objet**,

L'objet est une association des données et des procédures (qu'on appelle alors méthodes) ; **1pt**

2- Les principes de la POO :

- a- Encapsulation : **0.25 pt**
- b- Classe : **0.25 pt**
- c- Héritage : **0.25 pt**
- d- Polymorphisme : **0.25 pt**

3- Autres langage orientée Objet : 1pt

- C++ : très utilisé
- Java
- C# : langage de Microsoft (appartient à .NET)
- Objective C : langage utilisé par Apple
- PHP : langage tries utilisé sur le Web

- 4- Quand une fonction ne retourne aucun résultat, sa déclaration est la suivante : **1pts**

Void nom_fonction (type variables) ;

- 5- Une variable de type pointeur se déclare à l'aide du type de l'objet pointé précédé du

symbole *

Type * nom_du_pointeur;

1pts

Exercice N 1 :

a. 2.5 pts

```
#include <iostream>
using namespace std ;
    main()
    { int i, n, som ;
      som = 0 ;
i = 0 ; /* ne pas oublier cette "initialisation" */
      while (i<4)
      { cout << "donnez un entier " ;
        cin >> n ;
        som += n ;
i++ ; /* ni cette "incrémentation" */
      }
      cout << "Somme : " << som ;
    }
```

b. 2.5 pts

```
#include <iostream>
using namespace std ;
    main()
    { int i, n, som ;
      som = 0 ;
i = 0 ; /* ne pas oublier cette "initialisation" */
      do
      { cout << "donnez un entier " ;
        cin >> n ;
        som += n ;
```

```

    i++ ; /* ni cette "incrémentaion" */
        }
while (i<4) ; /* attention, ici, toujours <4 */
    cout << "Somme : " << som ;
        }

```

Exercice N 2 :

/* ----- Déclaration de la classe point ----- */ 3 pts

```

class Point
{
    /* déclaration des membres privés */

    private:
        float x,y ;

    /* déclaration des membres publics */
    public :
        point (float,float) ;
        void deplace (float, float) ;
        void affiche () ;
        } ;

```

Sauvegarde du fichier point1.h

/* ----- Définition des fonctions membres de la classe point ----- */ 4 pts

```

#include <iostream>

```

```

#include "point1.h"

```

```

Using namespace std

```

```

void Point : : point(float abs a, float ord)

```

```

        {
            x = abs;
            y = ord;
        }
void Point: : deplace(float dx, float dy)

        {
            x = x+dx;
            y = y+dy;
        }

void Point: : affiche() {
cout << " x = " << x << endl;
cout << " y = " << y << endl;
}

```

Sauvegarde du fichier point1.cpp

/* ----- utilisation de la classe point ----- */ 3 pts

```
# include <iostream>
```

```
# include "point1.h"
```

```
Using namespace std
```

```
main()
```

```

    {
        Point m, n ;
        point p(5, 2) ;
        p. affiche() ;
        p. deplace(8, 4) ;
        p. affiche() ;
    }

```

Sauvegarde du fichier exemple.cpp