

Exercice 1 : Questions de cours (1+0.5x5+1+1+1=6 points)

1. Donner l'extension(s) d'un fichier source C++
2. Définir les concepts suivant : Programmation orientée objet, Objet, Classe, polymorphisme, héritage
3. Donner la signification des mots clés suivant : Virtual ; Private.
4. Quelle est la sortie du code suivant? (1.5 points)

```
#include <iostream>
using namespace std;
class calculer
{
    int x, y;

public:
    void val(int , int);
    int somme()
    {
        return (x + y);
    }
};
void calculer::val(int a, int b)
{
    x = a;
    y = b;
}
int main()
{
    calculer calculer;
    calculer.val(5, 10);
    cout << "La somme = " << calculer.somme();
    return 0;
}
```

- (a) La somme = 5
- (b) La somme = 10

- (c) La somme = 15
- (d) Erreur parce que calculer est utilisé comme nom de classe et nom de variable dans la ligne 19.

Exercice 2 (choisir la(les) bonne(s) reponse(s)) (0.25x8=2pts)

1. Shape*fig; En supposant que le code ci-dessus est valide, on dit que fig est deShape.
 - (a) une variable / la classe
 - (b) une variable / l'instance
 - (c) une instance / la classe
2. Shape*fig = new Square(42); Ici, on dit que lede fig est Square *et que sonest Shape*.
 - (a) type statique / type dynamique
 - (b) type dynamique / type statique
3. Le code de la question précédente implique quedérive de.....
 - (a) Shape/Square
 - (b) Square/Shape
4. Quelle syntaxe permet de modifier la cible d'un pointeur p?
 - (a) *p = a;
 - (b) p = a; c)&p = a;
 - (c) impossible
5. Dans une classe, les constructeurs de ses objets membres sont appelées le code de son constructeur
 - (a) Avant
 - (b) après
 - (c) ca dépend du compilateur

6. Dans une classe, les destructeurs de ses objets membres sont appelés le code de son destructeur.
 - (a) Avant
 - (b) après
 - (c) dépend du compilateur
7. En C++, la déclaration d'une classe :
 - (a) peut contenir 0 ou plusieurs constructeurs
 - (b) doit toujours contenir 2 constructeurs
 - (c) ne doit contenir qu'1 seul constructeur
 - (d) doit contenir au moins 1 constructeur
8. En C++, on distingue des méthodes surchargées en fonction :
 - (a) de leurs noms
 - (b) uniquement du nombre de leurs paramètres
 - (c) de leurs types de retour
 - (d) du nombre ou du type de leurs paramètres

Problème:

Réalisation d'une classe point permettant de manipuler un point d'un plan. Voici un exemple de fichier d'en-tête **Point.h** qui décrit un objet qui représente un point dans l'espace euclidien bidimensionnel :

```
// A 2D point class
class point
{
private:
    // Declaration des proprietes d'un point ici.

public:
    // Constructeurs:
    // Constructeur par default ici.
```

```

// Constructeur a deux arguments x et y ici.

// Accessor methods
double get_x();
double get_y();

// Mutator methods
void set_x(double val);
void set_y(double val);
};

```

1. Ajoutez les propriétés privées de cette classe sachant que un point est défini par ses coordonnées x et y (des membres privés)
2. Ajoutez un constructeur par défaut et un constructeur paramétré.
3. Ajoutez une fonction membre déplace effectuant une translation définie par ses deux arguments dx et dy (double)
4. Ajoutez une fonction membre affiche se contentant d'afficher les coordonnées cartésiennes du point.
5. Ajoutez une fonction membre saisir se contentant de saisir les coordonnées cartésiennes du point.
6. Ajoutez une fonction membre distance effectuant calculant la distance entre deux point.
7. Ajoutez une fonction membre milieu donnant le milieu d'un segment.
8. Ecrivez un petit programme d'essai (main) gérant la classe point.