# 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;  
  }
  1. La somme = 5
  2. La somme = 10
  3. La somme = 15
  4. 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 de Shape.
   1. une variable / la classe
   2. une variable / l’instance
   3. une instance / la classe
2. Shape\*fig = new Square(42); Ici, on dit que le de fig est Square \*et que son est Shape\*.
   1. type statique / type dynamique
   2. type dynamique / type statique
3. Le code de la question précédente implique que dérive de.
   1. Shape/Square
   2. Square/Shape
4. Quelle syntaxe permet de modifier la cible d’un pointeur p?
   1. \*p = a;
   2. p = a; c)&p = a;
   3. impossible
5. Dans une classe, les constructeurs de ses objets membres sont appelées le code de son constructeur
   1. Avant
   2. après
   3. ca dépend du compilateur
6. Dans une classe, les destructeurs de ses objets membres sont appelés le code de son destructeur.
   1. Avant
   2. après
   3. dépend du compilateur
7. En C++, la déclaration d’une classe :
   1. peut contenir 0 ou plusieurs constructeurs
   2. doit toujours contenir 2 constructeurs
   3. ne doit contenir qu’1 seul constructeur
   4. doit contenir au moins 1 constructeur
8. En C++, on distingue des méthodes surchargées en fonction :
   1. de leurs noms
   2. uniquement du nombre de leurs paramètres
   3. de leurs types de retour
   4. 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 defaut 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 propriétes privées de cette classe sachant que un point est défini par ses coordonnées x et y (des membres privés)
2. Ajoutez un constructeur par défault 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 une fonction membre milieu donnant le milieu d’un segment.
8. Ecrivez un petit programme d’essai (main) gérant la classe point.