

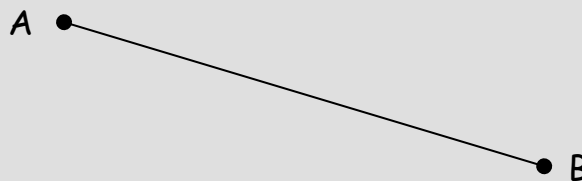
**EXAMEN DE FIN DU MODULE, (A distance)**  
**PROGRAMMATION ORIENTEE OBJETS EN LANGAGE C++**  
**Durée 1 h 30'**

**PROBLEME N° 1 : CONSTRUCTION ET EXPLOITATION D'UNE CLASSE « SEGMENT » :**

**(Partie du cours assurée en Présentiel)**

**Préliminaire : Rappel mathématique :**

Un segment est un ensemble de point linéaires.



Un segment est parfaitement connu sur le plan, si on connaît les coordonnées de deux points extrémités  $A(x, y)$  et  $B(z, t)$ .

L'objectif de ce travail est de construire une classe « **segment** » et de manipuler des objets créés à partir de cette classe.

**Travail N° 1 :**

*Créer un dossier, portant votre classe et votre nom, sur le bureau de la machine sur laquelle vous travaillez (Exemple : GLSID1\_Alamy --- ou --- BDCC1\_Naciri)*

**N.B. Très Très important :**

**Chaque chose que vous créez dans les programmes suivants, doit porter votre nom de famille. Exemples : pour M. HASSNAOUI, la classe « Segment » doit s'appeler obligatoirement « Hassnaoui\_Segment », la méthode « Définir » de travail N° 2 doit s'appeler obligatoirement « Hassnaoui\_Définir », ... etc.**

**Travail N° 2 :**

**2.1. Ecrire une classe nommée « segment », constituée de :**

- Quatre membres donnés  $x, y$  : coordonnées du point première extrémité et  $z, t$  : coordonnées du point deuxième extrémité du segment que l'on souhaite représenter,
- Quatre méthodes :
  - **Définir** : permettant de définir un segment à partir des coordonnées de ses deux extrémités,
  - **Longueur** : permettant de calculer la longueur du segment,
  - **Affiche** : permettant d'afficher les coordonnées des deux extrémités du segment à l'écran.

**2.2. Prévoir dans la fonction main(), des appels aux fonctions membres pour définir, calculer la longueur et afficher les deux extrémités du segment plusieurs fois.**

**2.3. Compiler, exécuter et enregistrer ce travail dans le dossier créé sous le nom « Votre nom\_Travail\_2.CPP ».**

### **Travail N° 3 :**

*Reprendre la classe « Segment » précédente et :*

**3.1. Proposer trois constructeurs différents de cette classe que vous écrivez séparément :**

- Le premier est une fonction sans paramètres permettant de définir un segment dont les coordonnées de sa première extrémité (0,0) et les coordonnées de sa deuxième extrémité (10,5),
- Le deuxième est une fonction avec deux paramètres permettant de définir un segment ayant les coordonnées de sa première extrémité tous égaux à la valeur du premier paramètre et les coordonnées de sa deuxième extrémité tous égaux à la valeur du deuxième paramètre.
- Le troisième est une fonction avec quatre paramètres permettant de définir un segment dont la première extrémité ayant comme abscisse la valeur du premier paramètre et comme ordonnée la valeur du second paramètre et dont deuxième extrémité ayant comme abscisse la valeur du troisième paramètre et comme ordonnée la valeur du quatrième paramètre.

**3.2. Ecrire un destructeur qui met fin à la vie de l'objet « Segment » en affichant les coordonnées de ses deux extrémités,**

**3.3. Compiler, exécuter et enregistrer ce travail dans le dossier créé sous le nom « Votre nom\_Travail\_3.CPP ».**

### **Travail N° 4 :**

*Reprendre la classe « Segment » précédente et :*

**4.1. Proposer une manière avec laquelle on peut à tout moment connaître et afficher le nombre d'objets « Segment » existants en mémoire,**

**4.2. Ecrire une fonction « Test » dans laquelle, vous créez deux objets S1 et S2 de classe « Segment »,**

**4.3. Ecrire la fonction main() dans laquelle, vous créez un objet S0, vous appelez ensuite la fonction « Test », puis vous créez un objet S3,**

**4.4. Compiler, exécuter, conclure et enregistrer ce travail dans le dossier créé sous le nom « Votre nom\_Travail\_4.CPP ».**

### **Travail N° 5 :**

*Reprendre la classe « Segment » précédente et :*

**5.1. Ecrire les deux fonctions membres surchargées suivante :**

- Une fonction : « **affiche()** » sans paramètres, permettant d'afficher les coordonnées des deux extrémités du segment,
- Une fonction : « **affiche(char \*)** » à un seul paramètre, une chaîne de caractères, permettant d'afficher à l'écran le nom que l'utilisateur désire donner au segment concerné et d'appeler la fonction : « **affiche()** » ci-dessus.

**5.2. Proposer une fonction main() créant des objets de cette dernière classe « Segment » et faisant appel à ces deux fonctions membres.**

**5.3. Compiler, exécuter et enregistrer ce travail dans le dossier créé sous le nom « Votre nom\_Travail\_5.CPP ».**

### **Travail N° 6 :**

*Reprendre la classe « Segment » la plus complète parmi toutes celles écrites ci-dessus :*

**6.1. En écrivant les différents constructeurs, destructeur et fonctions membres sous forme de fonctions « inline »,**

**6.2. Compiler et comparer la taille du fichier objet « .obj » du travail 6 à celui du travail de la classe que vous avez choisie,**

**6.3. Enregistrer ce travail dans le dossier créé sous le nom :  
« Votre nom\_Travail\_6.CPP ».**

**6.4. Commenter la différence des tailles des fichiers objets de la question 6.2. dans un fichier « Word » que vous enregistrez dans le dossier créé sous le nom : « Votre nom\_Travail\_6.DOC ».**

### **Travail N° 7 :**

**Reprendre la classe « Segment » précédente et :**

- 6.1. Introduire une fonction membre nommée « meme ». Cette fonction permet de détecter la coïncidence éventuelle entre deux segments et qui a comme paramètre un objet de classe « Segment ».**
- 6.2. Ecrire la fonction main() dans laquelle vous créez des objets et vous utilisez la fonction membre « meme » plusieurs fois,**
- 6.3. Compiler, exécuter et enregistrer ce travail dans le dossier créé sous le nom « Votre nom\_Travail\_7.CPP ».**

## **PROBLEME N° 2 : GESTION DES PRIX DANS UN MAGASIN**

**(Partie du cours assurée à Distance)**

**(BONUS sur 5 points en plus de la note du PROBLEME 1)**

### **Travail N° 8 (à faire dans le même dossier créé dans le travail N° 1) :**

Dans un magasin, un produit en vente est connu par les caractéristiques suivantes :

- Référence : entier
- Désignation : Chaîne de caractères
- Prix unitaire : Réel

**8.1. Elaborer une classe de base « Produit », formée des champs privés ci-dessus et des méthodes suivantes :**

- **ToString()** permettant d'afficher toutes les caractéristiques d'un produit,
- **GetPrix()** permettant de retourner le prix du produit,
- **SetPrix(float)** permettant de changer le prix du produit,
- **AfficherP()** permettant d'afficher le prix du produit,

Le magasin désire mettre certains produits en solde.

**8.2. Elaborer une classe « ProduitEnSolde », dérivée de la classe « Produit ».**

Cette sous-classe comprend une information additionnelle :

- **TauxRemise**, qui représente le pourcentage de réduction sur le prix d'origine du produit.

Et les méthodes suivantes :

- **GetRemise()** permettant de retourner le taux de remise du produit,
- **SetRemise(int)** permettant de changer le taux de remise.

Cette nouvelle classe doit redéfinir la méthode **GetPrix()**, afin de tenir compte de la remise sur le produit.

Elle va également redéfinir la méthode **AfficherP()**, afin que l'affichage donne également le Pourcentage de remise sur le prix d'origine.

**8.3. Implémenter les deux classes « Produit » et « ProduitEnSolde » dans un programme principal « main », en déclarant plusieurs instances de ces deux classes et en manipulant les différentes méthodes citées ci-dessus,**

**8.4. Compiler, exécuter et enregistrer ce travail dans le dossier créé sous le nom « Votre nom\_Travail\_8.CPP ».**

### **Travail N° 9 :**

- 9.1. Supprimer tous les fichiers exécutables de votre dossier, et ne laisser que les « .CPP »,**
- 9.2. Compresser votre dossier de travail en utilisant Winrar ou Winzip et envoyer le par mail à l'adresse « [mansouri.examens@gmail.com](mailto:mansouri.examens@gmail.com) » dans les 5 min qui suivent le temps de l'examen.**