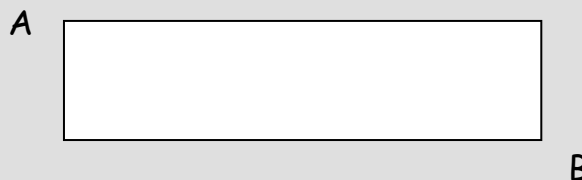


**EXAMEN DE FIN DU MODULE (A distance)**  
**PROGRAMMATION ORIENTEE OBJETS EN LANGAGE C++**  
**Durée : 2 h 30'**

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

**Préliminaire : Rappel mathématique :**

Un rectangle est une figure géométrique composée de quatre segments de droite parallèles deux à deux.



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

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

**Travail N° 1 :**

*Créer un dossier, portant votre classe et votre nom de famille, sur le bureau de la machine sur laquelle vous travaillez.*

**Exemples :**

- Un étudiant de la classe GLSID 1, qui s'appelle Slimani, va appeler son dossier : « GLSID1\_Slimani »,
- Un étudiant de la classe BDCC 1, qui s'appelle Alouane, va appeler son dossier : « BDCC1\_Alouane ».

**N.B.**

**Très --- Très important :**

**CHAQUE CHOSE** quelconque, que vous créez dans les programmes suivants, doit obligatoirement porter votre nom de famille.

**Exemples : pour M. Slimani :**

- la classe « Rectangle » doit s'appeler obligatoirement « Slimani \_ Rectangle »,
- la classe « Personne » du travail N° 8 doit s'appeler obligatoirement « Slimani \_ Personne »,
- la méthode « Définir » du travail N° 2 doit s'appeler obligatoirement « Slimani \_ Définir »,
- ... etc.

## **Travail N° 2 :**

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

- Quatre membres donnés x, y : coordonnées du premier point diagonal et z, t : coordonnées du deuxième point diagonal du rectangle que l'on souhaite représenter,
- Quatre méthodes :
  - **Définir** : permettant de définir un rectangle à partir des coordonnées de ses deux points diagonaux,
  - **Périmètre** : permettant de calculer le périmètre du rectangle,
  - **Surface** : permettant de calculer la surface du rectangle,
  - **Affiche** : permettant d'afficher les quatre sommets du rectangle à l'écran.

### ***2.2. Prévoir dans la fonction main(), des appels aux fonctions membres pour définir, calculer le périmètre, la surface et afficher les quatre sommets du rectangle 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 « Rectangle » 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 rectangle dont les coordonnées de son premier point diagonal (0,0) et les coordonnées de son deuxième point diagonal (10,5),
- Le deuxième est une fonction avec deux paramètres permettant de définir un rectangle ayant les coordonnées de son premier point diagonal tous égaux à la valeur du premier paramètre et les coordonnées de son deuxième point diagonal tous égaux à la valeur du deuxième paramètre.
- Le troisième est une fonction avec quatre paramètres permettant de définir un rectangle dont le premier point diagonal ayant comme abscisse la valeur du premier paramètre et comme ordonnée la valeur du second paramètre et dont le deuxième point diagonal 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 « Rectangle » en affichant les coordonnées de ses deux points diagonaux,***

#### ***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 « Rectangle » précédente et :***

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

#### ***4.2. Ecrire une fonction « Test » dans laquelle, vous créez deux objets R1 et R2 de classe « Rectangle »,***

#### ***4.3. Ecrire la fonction main() dans laquelle, vous créez un objet R0, vous appelez ensuite la fonction « Test », puis vous créez un objet R3,***

#### ***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 « Rectangle » précédente et :***

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

- Une fonction : « affiche() » sans paramètres, permettant d'afficher les coordonnées des deux points diagonaux à l'écran,
  - Une fonction : « **affiche(char \*)** » à un seul paramètre, une chaîne de caractères, permettant d'afficher à l'écran le nom du rectangle concerné et d'appeler la fonction : « **affiche()** » ci-dessus.
- 5.2. *Proposer une fonction main() créant des objets de cette dernière classe « Rectangle » 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 « Rectangle » 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.DOCX** ».*

### **Travail N° 7 :**

*Reprendre la classe « Rectangle » précédente et :*

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

## **PROBLEME N° 2 : GESTION D'UNE ECOLE**

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

On désire développer une application en langage C++, permettant de gérer les études dans une école supérieure. On suppose que cette opération fait intervenir des personnes de types étudiants et enseignants.

Une personne est connu par :

- Un nom : **Nom**, chaîne de caractères,
- Un prénom : **Prenom**, chaîne de caractères,
- Une date de naissance : **Date**, chaîne de caractères,
- Une méthode : **Afficher()**, permettant d'afficher toutes les caractéristiques d'une personne,

Un étudiant est connu, **en plus**, par :

- Un niveau d'étude : **Niveau**, entier,
- Une spécialité : **Specialite**, chaîne de caractères,

- Des notes : **Note**, un tableau de notes, sachant qu'un étudiant passe 10 matières à l'examen,
- Une méthode : **Saisie()**, permettant la saisie des données d'un étudiant,
- Une méthode : **Afficher()**, permettant d'afficher, à l'écran, les données d'un étudiant,
- Une méthode : **Moyenne()**, permettant de retourner comme résultat la moyenne des notes d'un étudiant,

Un enseignant est connu, **en plus**, par :

- Un diplôme : **Diplôme**, Chaîne de caractère,
- Une spécialité : **Specialite**, chaîne de caractères,
- Un nombre d'heure : **Nombre\_Heures**, un réel, qui représente le nombre d'heures travaillées dans le mois,
- Un taux horaire : **Taux**, un entier, qui représente le prix de l'heure travaillée par l'enseignant,
- Une méthode : **Saisie()**, permettant la saisie des données d'un enseignant,
- Une méthode : **Afficher()**, permettant d'afficher toutes les données d'un enseignant,
- Une méthode : **salaire()**, permettant de retourner comme résultat le salaire d'un enseignant, calculé en fonction du nombre d'heures travaillées dans le mois et du taux horaire, diminué de 17% pour les impôts.

- 8.1. *Ecrire les trois classes « Personne », « Etudiant » et « Enseignant ». On devra utiliser, pour cela, le principe de l'héritage.*
- 8.2. *Ecrire une fonction « Presenter », ayant comme argument un objet personne, et faisant appel aux différentes fonctions « afficher() » citées ci-dessus. Ces dernières doivent être **polymorphes**, pour permettre à la fonction « Presenter » de reconnaître l'origine de la personne : Etudiant ou enseignant,*
- 8.3. *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 » Compresser votre dossier de travail en utilisant Winrar ou Winzip et envoyer le par mail à l'adresse « [poocpp2021@gmail.com](mailto:poocpp2021@gmail.com)» dans les 5 min qui suivent le temps fin de l'examen.***