

1. Objectif de la séance

Le but de la séance est de manipuler les notions suivantes :

- Les constructeurs, mutateurs et accesseurs
- La création des objets dynamiques
- Les pointeurs, références et tableaux
- Passage de paramètres

2. Notions du pointeur, référence et tableau

Question 1 : Donnez les valeurs de A,B,C, P1 et P2 suite à ce main

```
int A =1, B=2, C=3 ;
int *P1, *P2 ;
P1=&A
P2=&C
*P1=(*P2)++
P1=P2
P2=&B
*P1--=*P2
++*P2
*P1*=*P2
A=++*P2**P1
P1=&A
*P2=*P1/=*P2
```

Question 2 : Un pointeur et une référence

- Déclarez un tableau d'entiers A de dimension 10
- Développez deux méthodes, `remplitA` et `afficherA`, permettant de remplir et d'afficher le tableau.
- Déclarez une référence vers l'élément 5 du tableau et incrémenter sa valeur
- Déclarez un pointeur vers l'élément 7, incrémenter le pointeur et la valeur pointée

Question 3 : Passage de paramètre par valeur vs Passage de paramètre par référence vs Passage de paramètre par pointeur

Voici trois implémentations de la méthode **square** :

```
int square(int n) {
    cout << "In square(): " << n << endl; // 0x22ff00
    n *= n;
    return n;
}
```

```
void square(int * pNumber) {
    cout << "In square(): " << *pNumber << endl; // 0x22ff1c
    *pNumber *= *pNumber;
}
```

```
void square(int & rNumber) { }
    cout << "In square(): " << rNumber << endl; // 0x22ff1c
    rNumber *= rNumber
}
```

Implémentez la méthode **main**, où vous appelez la méthode **square**. Appelez la méthode **cout** pour afficher la valeur de l'entier passé en paramètre. Quand est-ce que la valeur du nombre passé en paramètre change ?

3. Gestion des données des étudiants

On veut développer une application qui permet de saisir les données des étudiants et de calculer leurs moyennes. On va commencer tout d'abord par la création d' :

- Une classe intitulée **Matiere** avec :
 - Attributs : intitulé de type **tableau de char**, coefficient de type **double**, note de type **double**
 - Méthodes :
 - Les accesseurs (getters & setters) (voir l'annexe pour comprendre c'est quoi)
 - Un constructeur
 - Un destructeur
- Une classe intitulée **Etudiant** avec :
 - Attributs : Num_carte de type **int**, Téléphone de type **int**, Moyenne de type **double**, matieres est un **Tableau de type Matiere**
 - Méthodes :
 - Les accesseurs (getters & setters)
 - Méthode **void afficher()** : permet d'afficher les données d'un étudiant
 - Méthode **double calcul_moyenne(Tableau de taille 2 de type Matiere)** : permet de calculer la moyenne de l'étudiant – utilisez la bibliothèque <cmath>

- Méthode **boolean reussi()** : permet de déduire si l'étudiant en question a réussi ou pas s'il aura une moyenne supérieure à 10
- Un constructeur
- Un destructeur
- Une classe intitulée **Filiere** avec :
 - Attributs : id_filiere de type **int**, Etudiants de type **Tableau d'Etudiant**
 - Méthodes :
 - Les accesseurs (getters & setters)
 - Méthode **void afficher()** : permet d'afficher les données de tous les étudiants de la filières
 - Un constructeur
 - Un destructeur

Les attributs seront privés et les méthodes publiques.

Travail demandé :

1. Ecrivez le code des classes demandées.
Rendez-vous à votre EDI. Créez un nouveau projet. Nommez le TP2_Gestion_Etudiant.

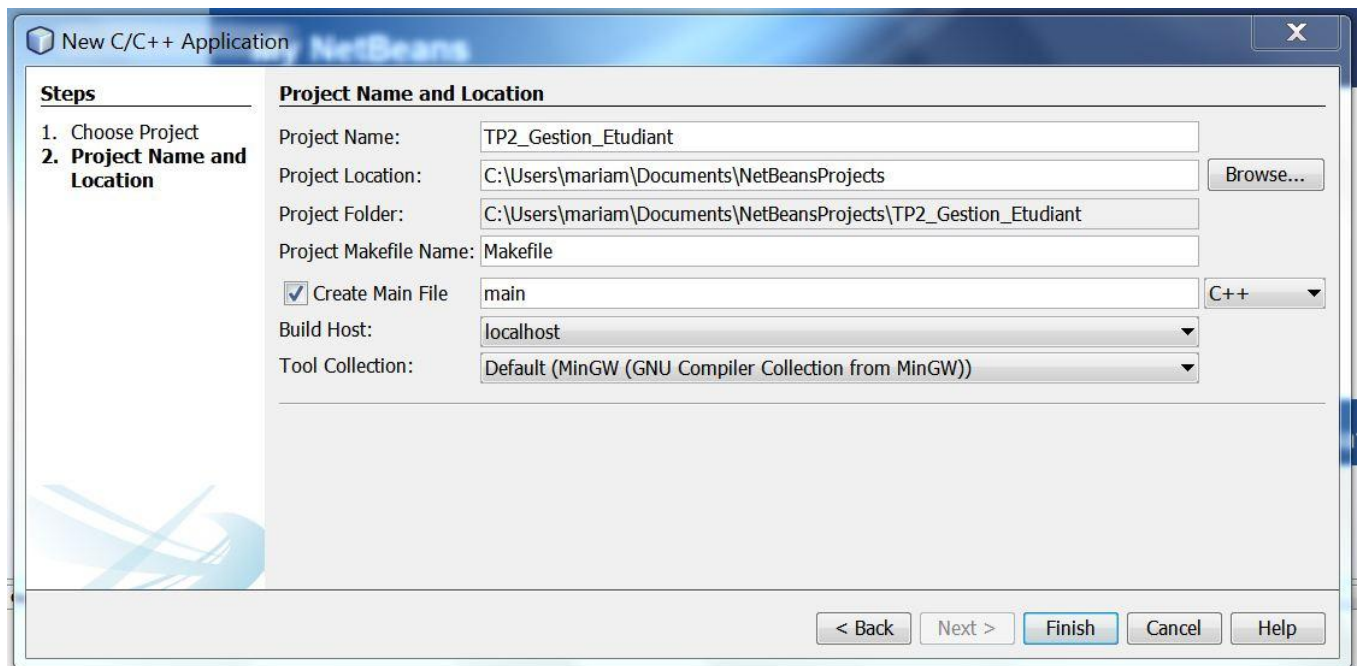


Figure 1 Créer un nouveau projet

Une fois que votre projet est créé, ajoutez un nouveau « C++ Main File » comme montre la figure 2. Commencez à écrire vos classes.

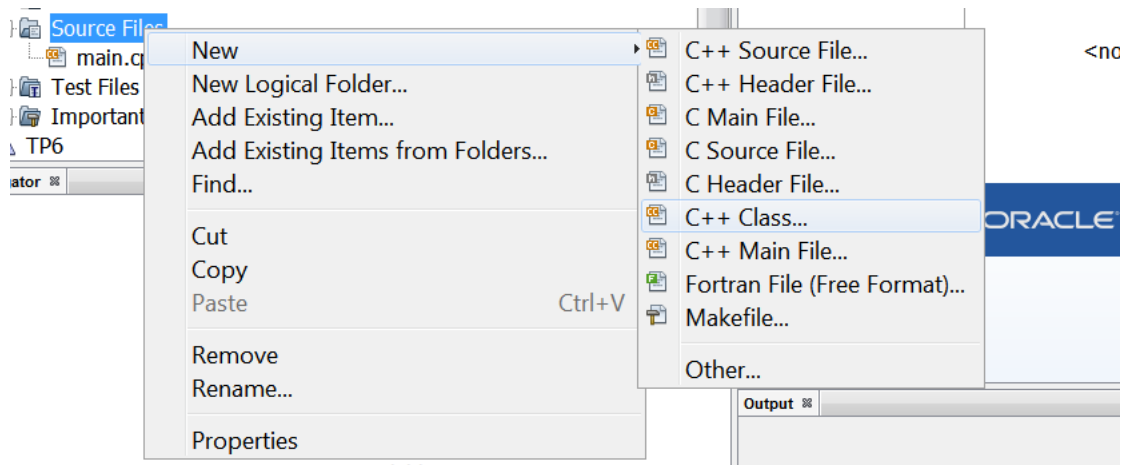


Figure 2 Créer une nouvelle classe C++

2. Ecrivez une méthode main pour tester le scénario suivant :
 - a. Vous pouvez choisir entre demander la saisie des données ou de créer les objets demandés sans passer par la saisie du clavier
 - b. Tester avec 4 étudiants, 3 matières et 2 filières
3. Surchargez un des constructeurs. Utilisez la liste d'initialisation
4. On veut ajouter une méthode intitulée **boolean appartenance(parametre)** à la classe appropriée (argumentez votre choix) pour vérifier si un Etudiant, passé en paramètre, appartient à une filière donnée.
On va rajouter 3 variantes de cette méthode. Insérez-les dans votre main. C'est quoi la différence ? Expliquez.

boolean appartenance(Etudiant e)
boolean appartenance(Etudiant *e)
boolean appartenance(Etudiant &e)

Annexe

C'est quoi un accesseur ou getter?

Un accesseur est une fonction membre permettant de récupérer le contenu d'une donnée membre protégée. Un accesseur, pour accomplir sa fonction¹ :

- doit avoir comme type de retour le type de la variable à renvoyer
- ne doit pas nécessairement posséder d'arguments

Une convention de nommage veut que l'on fasse commencer de façon préférentielle le nom de l'accesseur par le préfixe *Get*, afin de faire ressortir sa fonction première.

¹ <http://www.commentcamarche.net/contents/183-les-accesseurs-et-les-mutateurs-en-langage-c>

```
class Personne{
    private :
        int age;
    public :
        int GetAge();
};
int Personne::GetAge(){
    return age;
}
```

C'est quoi un mutateur ou setter?

Un mutateur est une fonction membre permettant de modifier le contenu d'une donnée membre protégée. Un mutateur, pour accomplir sa fonction¹ :

- doit avoir comme paramètre la valeur à assigner à la donnée membre. Le paramètre doit donc être du type de la donnée membre
- ne doit pas nécessairement renvoyer de valeur (il possède dans sa plus simple expression le type void)

Une convention de nommage veut que l'on fasse commencer de façon préférentielle le nom du mutateur par le préfixe Set.

```
class Personne{
    private :
        int _age;
    public :
        void SetAge(int);
};
void Personne::SetAge(int age){
    _age = age;
}
```