

# Programme de soutien en C++



# Développement d'une bibliothèque C++ pour les étudiants de l'IUT d'Orléans.

## Constat :

Un certain nombre de fonctionnalités (check de saisie, création de grille, débogage, smartPtr, singleton ...) vous seront utiles tout au long de votre cursus Universitaire (dans tous les cas pour le DUT).

## Idée générale du soutien :

- Eviter de faire des copier/coller entre projets, ne pas être dépendant d'une bibliothèque pouvant changer du jour au lendemain.
- Néanmoins, il ne faut pas réinventer la roue : dans la mesure du possible, nous étudierons comment profiter de la librairie standard (STL).
- Avoir une approche différente de la programmation que celle qui a pu vous être donnée par vos enseignants. L'idée étant ici de concevoir un outil qui complétera ceux que vous utilisez déjà pour la conception de vos projets.

## Etape 1 : Prise en main d'un IDE avec Eclipse CDT

Un IDE (environnement de développement intégré) est un programme regroupant un ensemble d'outils pour le développement de logiciels.

Cela permet une meilleure organisation de vos projets, de faciliter la compilation (détection automatique des erreurs de syntaxe/logique) etc. Il en résulte donc un gain de temps considérable.

Le choix d'Eclipse est simple : il est libre, gratuit et multiplateforme. De plus, bien qu'Eclipse soit, à la base, conçu pour JAVA, le plugin CDT est relativement performant pour programmer en C++.

Enfin, il vous sera utile en deuxième année dans le cadre de vos cours en JAVA.

### **1) Installation et configuration :**

- ⇒ Téléchargement et installation d'Eclipse CDT sous Windows (MinGW : <http://tdm-gcc.tdragon.net/>) /Linux
- ⇒ Aperçu rapide du fonctionnement de l'IDE (affichage des lignes, jonglage entre les vues, menu projet (gestion propriétés du projet), menu Windows (gestion propriétés Eclipse), ...)
- ⇒ Création d'un nouveau projet (définition d'un style de code Windows=>Préférence=>C/C++=>Style code)
- ⇒ Configuration de base pour la compilation (compilation auto, etc.)

## 2) Apprentissage des raccourcis Eclipse :

⇒ Création d'une classe Chaine (utilisant les char \* pour Coplien) afin de donner un contexte à l'utilisation des raccourcis.

1. Ctrl+Shift+F : Formater le code (sélectionné ou tout le fichier)
2. Ctrl+Shift+/ : Commenter/Décommenter le bloc sélectionné
3. Ctrl+Shift+G : Rechercher les références à l'élément sélectionné
4. Alt+Shift+R : Renommer
5. Ctrl-F11/F11 : Exécution en mode normal/débogage
6. F3 : Open déclaration
7. F4 : Hiérarchie d'une classe! (Ctrl O : Hiérarchie d'un fichier)
8. Ctrl+Space : Auto complétion
9. Ctrl+Shift+L : Liste des raccourcis

Tous les raccourcis sont, bien entendu, configurables (voir : Windows=>Préférence=>General=>Key ou Ctrl+Shift+I+I).

## 3) Prise en Main du débogueur :

⇒ Exécution du code précédent (jeu de test sur la classe Chaine) pas à pas afin de voir le comportement du programme.

### Etape 2 : La conception de la bibliothèque (Nommé Aumiis)

La question de départ qu'il faut se poser avant de se lancer dans un développement est : de quoi avez-vous besoin ? Par exemple :

- Lors d'un débogage, nous aimerions éviter d'écrire ce genre de code :

```
int entier = 0;
std::cout<<"La valeur de la variable entier est : ("<<entier << ")"<<std::endl;
```

- Faciliter le stockage de données.
- Faciliter la gestion des pointeurs.
- ...

## 1) Création du projet et configuration : *Parcours en détail des propriétés du projet*

## 2) Debug\_hpp : *Regroupe un ensemble de code permettant de faire du débogage en mode texte.*

⇒ Rappel sur les commandes préprocesseur (<http://www.cppreference.com/wiki/fr/preprocessor/start>).

⇒ Rappel sur les Templates (pour la fonction toString()).

**3) Conteneur\_hpp :** Permet la création de grilles de données et de vectors.

- ⇒ Conception d'une classe vector ressemblant à celle de la STL (vérification du niveau des étudiants).
- ⇒ Aperçu rapide de la STL (<http://www.cppreference.com/wiki/>).
- ⇒ Aperçu rapide sur le fonctionnement des vectors de la STL (+iterator) à travers un jeu de tests.  
+ Explication des différences (en temps d'exécution) entre les tableaux, les listes chaînées et les arbres de recherche.
- ⇒ Rappel sur le fonctionnement des flux de la STL  
(<https://judge.inf.ethz.ch/doc/cpp/reference/iostream/index.html>)
- ⇒ Retour sur DEBUG\_HPP afin de faire un opérateur de flux sur les vectors (utilisation de std::copy).
- ⇒ Codage d'une grille utilisant les vectors de la STL.

**4) Direction\_hpp :** Création d'un système de direction pour se déplacer dans la grille.**5) Fonctionnal\_HPP :** Création d'un générateur de cartes aléatoires (utilisant la grille) et implémentation d'un plus court chemin sur cette grille.**6) SmartPtr\_hpp :**

- ⇒ Explication du principe.
- ⇒ Un pointeur avec compteur de références : shared\_ptr.
- ⇒ Un pointeur gérant la copie des données : copy\_ptr.
- ⇒ Présentation de ce que propose la STL (et de ce qu'elle va proposer en c++0x).

**Etape 3 : Vers les éclipses et au-delà ...****1) L'utilisation d'Aumiis en temps que bibliothèque (linkage) :**

- ⇒ Dynamique : (Windows : .dll / Linux : .so) Compilation => Appelle les fonctions des DLL/SO. Mais au lieu que les fonctions soient incluses dans l'exécutable, elles sont chargées lors de l'exécution (Exécutable plus léger, consommation de moins de mémoire mais il faut avoir la dll/SO pour exécuter le programme).
- ⇒ Statique : (Windows : .lib / Linux : .a) Comme pour la dll, compilation, mais au lieu de charger les fonctions à l'exécution, celle-ci sont chargées lors de l'édition des liens.

## 2) Utilisation des Templates d'Eclipse :

- ⇒ Création de doc, de boucles ...

## 3) Gestion des projets : *Utilisation des WorkingSet*

## 4) Installation et utilisation de plugins :

- ⇒ Utilisation de SVN avec Eclipse
- ⇒ Utilisation de Doxygen avec Eclipse grâce à Eclox

## Etape 4 : A vous de jouer !

- ⇒ Le code proposé définit les fonctionnalités de base. Des idées d'améliorations se trouvent en commentaire à la fin de chaque fichier de définition.
- ⇒ Plus que des modifications, des réels ajouts à votre librairie
  - Gestion réseau
  - Gestion saisie
  - Singleton ...
  - Vérificateur de fuite de mémoire
  - Etc.