

TRAVAUX PRATIQUES 1

Ce TP porte sur le chapitre « Les Bases du C++ ».

1 Prise en main de Code::Blocks 17.12

Exercice 1:

Fin de configuration de l'IDE

À partir des "postes pédagogiques" (salle 320B et 304B), vous avez accès à un espace partagé entre toutes les machines de l'université. Ce disque commun est monté par défaut sur les machines Windows :

- En tant que disque S:\ (nommé "pg-srv3" + votre login) sur chaque machine, accessible depuis l'explorateur sous la catégorie "Ordinateur".
- En tant que répertoire "Mes Documents", accessible depuis l'explorateur sous la catégorie "Bibliothèques" puis "Documents".

Ces emplacements sont strictement équivalents.

- ◇ Créer un fichier texte "test" dans l'un (clic droit, "Nouveau", "Document texte") et vérifier qu'il apparaît bien dans l'autre.

Code::Blocks gère chaque "application" que vous développez en tant que "projet" : ensemble des fichiers source et en-tête de l'application, auxquels s'ajoutent les fichiers de compilation et un fichier de configuration, d'extension ".cbp". Ce projet est contenu dans un répertoire du nom de l'application.

Pour faciliter votre activité, nous allons créer un répertoire dans votre espace personnel où vous enregistrerez systématiquement vos projets Code::Blocks.

- ◇ Créer le dossier S:\CodeBlocks.

Ouvrir maintenant le programme Code::Blocks, il peut éventuellement (si c'est le premier lancement sur cette machine) vous demander s'il doit associer les fichiers C++ à l'application et/ou vous demander de valider les compilateurs.

- ◇ Valider l'association des fichiers C++ à l'application : "Yes, associate Code::Blocks with C/C++ file types"
- ◇ Vérifier la présence du compilateur "GNU GCC Compiler" (ou "MinGW Compiler").
- ◇ Valider son statut de compilateur "par défaut".

Exercice 2:

Premiers pas dans Code::Blocks

Vous aller maintenant créer votre première application C++.

- ◇ Ouvrir le menu "File"
- ◇ Choisir "New" puis "Project..."

Vous avez lancé l'assistant de création de projet. Nous allons créer une simple application affichant du texte dans la console.

- ◇ Chercher le type "Console application" (Categorie "Console") et le sélectionner
- ◇ Sélectionner le bouton "Go"

Les écrans suivants vont permettre de décider du nom de l'application et du lieu d'enregistrement des fichiers.

- ◇ Sélectionner "Next>"
- ◇ Sélectionner "C++" puis "Next>"
- ◇ Le "Project title" définit le nom de l'application, choisir un nom idéalement **sans espace ni accent**.
- ◇ Pour le dossier d'enregistrement "Folder to create project in", cliquer le bouton "...", sélectionner le dossier S:\CodeBlocks (depuis "Mes Documents" ou "Ordinateur") et valider.
- ◇ Les autres champs sont complétés automatiquement, sélectionner "Next>"
- ◇ Laisser les options de bases concernant les cibles de compilation, sélectionner "Finish"

Votre espace de travail s'ouvre sur votre application toute neuve !

Le panneau de gauche ("Management") permet de naviguer dans votre code, soit sous forme de "Project" (la vue de base et celle qu'on utilisera le plus souvent) soit en triant les "Symboles" (fonctions, espace de noms, variables, classes, ...)

- ◇ Dérouler l'arborescence "Sources"
- ◇ Double cliquer le fichier "main.cpp"

Le panneau central principal ouvre le fichier "main.cpp" et vous permet de l'éditer. L'assistant de Code::Blocks vous a généré une trame de base simple mais complète.

Il ne reste plus qu'à compiler l'application.

- ◇ Ouvrir le menu "Build"
- ◇ Sélectionner "Build" pour compiler l'application

Le panneau "Log & others" en bas de la fenêtre d'édition permet de visualiser les informations retournées par les différents outils de l'IDE, elle affiche à ce moment le rapport de compilation dans "Build log". Vous pouvez y lire la ligne de compilation exécutée, la taille de l'exécutable, le temps de compilation...

L'onglet "Build messages" présente plus simplement les messages envoyés lors de la compilation.

Vous pouvez augmenter la taille de ce panneau "Log & others" en tirant sa bordure supérieure vers le haut.

- ◇ Vérifier la présence de l'option `-std=c++14` dans la ligne de compilation de l'onglet "Build messages"
- ◇ En son absence, suivre les instructions de l'enseignant

Nous allons maintenant introduire une erreur de compilation et observer le comportement de l'IDE.

- ◇ Dans la fenêtre d'édition, effacer la ligne 3 `using namespace std;`
- ◇ Sauvegarder le fichier (menu "File", option "Save file")
- ◇ Recompiler
- ◇ Observer les messages d'erreur dans l'onglet "Build messages".

La fenêtre d'édition affiche un indicateur rouge pour chaque ligne où le compilateur estime qu'il existe une erreur, ces erreurs apparaissent en rouge dans l'onglet "Build messages" et un clic sur ces lignes rouges vous positionne directement sur la ligne concernée. L'onglet "Build log" indique le message d'erreur complet, qui peut apporter des informations complémentaires.

Les messages en bleu indiquent des alertes, on peut aussi cliquer dessus pour atteindre la ligne concernée.

- ◇ Rectifier le code en rajoutant la ligne effacée
- ◇ Sauvegarder et recompiler
- ◇ Lancer l'application : menu "Build", option "Run"

Une fenêtre (la fameuse "console" mentionnée dans l'assistant) s'ouvre affichant le texte "Hello World!". Notez les informations supplémentaires ajoutées par Code::Blocks et non mentionnées dans le code C++.

Les menus de l'IDE indiquent – en complément des options que vous avez choisies – des représentations sous forme d'icônes de ces actions. Ces icônes apparaissent dans les barres d'outils de l'IDE, permettant de lancer ces actions directement.

De même des raccourcis clavier sont souvent mentionnés, pour le même effet.

Exercice 3:

Reprendre le second exemple de code du cours "Racines carrées" (support diapo 1.22).

- (a) Recopier les instructions en essayant de comprendre et définir le comportement de chacune des lignes. En cas de doute, se référer au "décodage" des diapos suivantes ou aux sections adéquates du cours.
- (b) Observer les aides fournies par l'IDE lors de la frappe :
 - i. Indentation : Vous pouvez insérer une indentation avec la touche TAB, Code::Blocks vous propose aussi une indentation automatique de deux façons :
 - Pour tout le fichier : Menu "Plugins", option "Source Code formatter (AStyle)"
 - Pour une partie du code : Sélectionner le texte, clic droit, option "format use AStyle"Sans sélection, la seconde option a le même effet que la première.
 - ii. Auto-complétion : un menu apparaît lors de la saisie d'identificateur que l'IDE reconnaît :
 - lors de la saisie, on peut naviguer dans les choix avec ↑ et ↓ et appuyer sur "entrée" pour valider, l'identificateur est alors complété
 - une fois la saisie interrompue, on peut appeler l'auto-complétion à la position actuelle du curseur par le menu "Edit", option "Complete code" (raccourci clavier : ctrl+espace)
 - iii. Coloration syntaxique : Elle aide la lecture et permet de confirmer l'existence d'un mot clef du langage
 - iv. Information : Lors de l'utilisation d'une fonction ou sa sélection pour auto-complétion, les informations connues du compilateur sur la fonction sélectionnée sont affichées : signature, argument en cours de saisie,...
 - v. Aide à la saisie : Plusieurs raccourcis clavier sont à votre disposition, entre autre ctrl+d pour effacer la ligne en cours et ctrl+l (L) pour dupliquer la ligne en cours¹. Naviguer dans les menus pour repérer les raccourcis les plus utiles. Ne pas hésiter à tester.
- (c) Compiler, si des erreurs de compilation apparaissent, essayer de les corriger sans l'aide du support de cours en déchiffrant le message du compilateur dans les onglets "Build messages" et "Log message".
- (d) Une fois les éventuelles erreurs corrigées, exécuter et observer le résultat.
- (e) Modifier le code en ajoutant les fonctions décrites dans la parties sur les "fonctions secondaires" (support diapo 1.34) puis en les utilisant dans le code.

2 Les bases du C++ : Les bases du langage

Exercice 4:

Reprendre l'exercice 4 du TD1, vérifier les résultats.

Exercice 5:

Reprendre l'exercice 5 du TD1, partie (b) en s'aidant du cours traitant de l'instruction "switch" (support diapo 1.66) pour ces deux cas :

1. Liste complète : http://wiki.codeblocks.org/index.php/Keyboard_Shortcuts

- (a) Indiquer la valeur des constantes "v" et "w" par une directive "#define"
- (b) Indiquer la valeur des constantes "v" et "w" avec une expression "constexpr"

Exercice 6:

En partant de l'exercice 6 du TD1 :

- (a) Écrire le code décrit en partie (c) avec un "if ... else"
- (b) Faire la partie (d) avec un "switch"
- (c) Constater le gain en terme d'écriture allégée et d'efficacité dans l'ajout d'un cas supplémentaire en introduisant 2 autres monnaies.

Exercice 7:

- (a) Coder la solution de l'exercice 7 du TD1
- (b) Extraire le calcul dans une fonction de prototype suivant : `int multiplication (int , int)`
- (c) Surcharger la fonction pour les types "double" et "float"

Exercice 8:

Faire les exercices 8, partie (a), puis 9 du TD1

3 Les bases du C++ : Utilisation avancée

Exercice 9:

À partir de l'exercice 6 du TD2 :

- (a) Écrire chaque variante de la fonction "`int traitement (...)`" présentée dans la partie (a) en prenant soin de :
 - modifier dans le corps de la fonction chaque argument
 - constater l'effet sur les variables passées en argument dans la portée principale (en affichant leur contenu avant et après appel de la fonction, par exemple)
 - adapter la fonction "main()" où sont déclarées les variables servant d'argument à chaque fois

Pour la suite, pensez à commenter vos versions précédentes plutôt que de les effacer..

- (b) Tester l'observation de la partie (b) de l'exercice, que se passe-t-il si les variantes sont définies mais non appelées dans la fonction "main()" ?

Exercice 10:

À partir des questions 8 et 9 du TD2 :

- (a) Faire les parties (b) et (c) de l'exercice 8, répondre à la question (c) en proposant un code (court mais complet) de démonstration.
- (b) Faire l'exercice 9 en écrivant un code comprenant une fonction "main()" et les deux "affiche ()"
- (c) Insérer la réponse de la question (c) de l'exercice 8 dans la boucle "main()" et dans les fonctions "affiche ()" de la question précédente.
Noter vos observations.
- (d) À l'aide de ces observations, terminer l'exercice 8.