

## Types Abstraits & Base de la POO - TD 1

† † †

### Environnement de développement pour le langage C

## I - Environnement de Développement

Exercice I.1

### IDE Visual Studio Code

VisualStudio Code est un IDE Microsoft, multi-plateformes, multi-langages. Il devrait être installé sur les machines de l'École. Suivez les étapes suivantes afin de configurer votre dossier de travail.

1. Créer un répertoire de travail sur votre disque nommé Workspace-tabpoo
2. Ouvrir VisualStudio Code et cliquer sur "Add workspace folder ...". Choisir le répertoire précédemment créé.
3. Dans l'explorer de l'IDE, créer un répertoire test-env à la racine de votre workspace.
4. Dans ce nouveau répertoire, créer un nouveau fichier un-test.c et installer le plug-in associé au langage C si nécessaire.
5. Recopier le code suivant.

```
1 #include <stdio.h>
2
3 int main (int argc, char *argv[]) {
4     printf("Coucou\n");
5     printf("les\n");
6     printf("gens de TABPOO !!\n");
7     return 0;
8 }
```

Listing 1 – "Fichier un-test.c"

## II - Toolchain GCC

Durant les activités en langage C de ce cours, nous utiliserons la *toolchain* GCC.



GCC :

L'exécutable gcc est un programme permettant d'appeler le préprocesseur, le compilateur, l'assembleur et le linker pour les langages C et C++. Les opérations réalisées sont les suivantes :

- appel du préprocesseur C (nommé cpp) ;
- appel du compilateur C (nommé cc1), ou C++ (cc1plus) selon le type de fichier à compiler (les fichiers .c sont compilés en C, les fichiers .C, .cc et .cpp sont compilés en C++). Les compilateurs génèrent un fichier assembleur ;
- appel de l'assembleur (as) pour générer le fichier objet, si l'option -S n'est pas spécifiée ;
- appel de l'éditeur de liens (ld) pour générer l'exécutable ou la bibliothèque, si l'option -c n'est pas spécifiée. Dans ce cas, la librairie C standard est incluse par défaut dans la ligne de commande de l'édition de liens.

Le programme gcc accepte différentes options. Voici quelques unes des options les plus fréquentes :

Option	Description
--help	Affiche l'aide de gcc.
--version	Donne la version de gcc.
-E	Appelle le préprocesseur. N'effectue pas la compilation.
-S	Appelle le préprocesseur et effectue la compilation.
-c	Appelle le préprocesseur, le compilateur et l'assembleur, mais pas l'éditeur de lien. Seuls les fichiers objets (.o) sont générés.
-o nom	Spécifie le nom du fichier objet généré.
-w	Supprime tous les warnings.
-W	Active les warnings supplémentaires.
-Wall	Active tous les warnings possibles.
-g	Construit une version exécutable par le debugger.

*Exercice II.1*

## Compilation & Construction

1. Dans la vue “terminal” de l’IDE, saisir la commande suivante : `gcc -o un-test.exe un-test.c`
2. Lister le contenu du répertoire et remarquer la création du nouvel exécutable
3. Lancer l’exécution par la commande : `./un-test.exe`

*Exercice II.2*

## Debuggage

Nous allons maintenant exécuter notre programme en mode “debug”.



### GCC - Debuggage :

Pour pouvoir utiliser le mode de debuggage proposé par GCC, il est nécessaire de reconstruire l’exécutable en utilisant l’option `-g` (Cf. tableau précédent). La toolchain GCC permet alors d’exécuter ce programme en permettant de nombreuses opérations durant l’exécution. Voici un petit résumé des actions les plus classiques :

Action	Description
<code>quit(q)</code>	quitter gdb
<code>run(r)</code>	lancer l’exécution
<code>start</code>	lancer l’exécution et arrêt au début de la procédure principale
<code>break,watch,clear,delete(b,w,cl,d)</code>	introduire un point d’arrêt, ou “surveiller” une variable, effacer un point d’arrêt
<code>info break (i b)</code>	affiche tous les break et watch
<code>step,next,until,continue(s,n,u,c)</code>	avancer d’un pas en entrant ou pas dans les sous-fonctions ou fin de boucle, continuer
<code>print,backtrace,list(p,bt,l)</code>	afficher la valeur d’une variable, la pile d’exécution, afficher le code où on est arrêté.

4. Construire le programme en mode “debug” : `gcc -g -o un-test.exe un-test.c`
5. Lancer le debugger sur le programme : `gdb un-test.exe`
6. Réaliser une exécution complète.
7. Réaliser une exécution “pas-à-pas”

‡ ‡ ‡