ISIE4 – LOO C++ - Fiche de séance (TD / AP)

TD1 – Premiers pas en C++

Référence/type[1]	LOO C++ - TD1		☒ TD ☐ AP
Thématique principale	Premiers pas en C++		
Thématique(s) secondaire(s)	✓ Prise en main environnement de développement – Compilation, production d'exécutable ✓ Déclaration / Définition / Initialisation de variables ✓ Déclaration / Définition de fonctions – Passage de paramètres ✓ Types structurés – Déclaration et utilisation ✓ Types structurés – Passage de paramètre par référence		
Durée	2h	***Niveau[2]***	A
Prérequis	✓ LOO C++ CM1		
Compétences opérationnelles et niveau cible[3]	✓ Compiler et créer un exécutable à partir d'un ou plusieurs fichiers source C++, de préférence à partir du compilateur GNU GCC, sous un OS GNU/Linux (***M***) ✓ Maîtriser le cycle de vie d'une variable ou d'un objet (***A***) ✓ Déclarer et appeler une fonction (***M***) ✓ Surcharger une fonction (***A***) ✓ Définir un type structuré, utiliser des objets de ce type (***M***) ✓ Maîtriser le passage de paramètres par référence pour les objets (***A***) ✓ Utiliser des types constants (***M***) ✓ Identifier l'intérêt et exploiter les expressions constantes (constexpr) (***A***)		
Modalités et critères d'évaluation	Auto Evaluation		
Matériel(s), équipement(s), composant(s) nécessaires			
Logiciel(s) nécessaire(s)[4]			
Ressources			

Avant la séance…

Revoir le CM1, notamment ce qui concerne les variables et les fonctions.

Anticiper l'utilisation de son propre environnement de développement C++ (dans le cas où l'environnement proposé dans la VM fournie ne conviendrait pas).

Remarque : Il existe aujourd'hui des « compilateurs » (entre guillemets car ces outils sont souvent bien plus que cela) en ligne, qui peuvent permettre de réaliser tout ou partie des exercices proposés dans le cadre de l'enseignement de C++. Il peut être intéressant de regarder un peu ces outils, on citera par exemple (il y en a d'autres) :

- ✓ *https://www.onlinegdb.com/online_c++_compiler : compilateur **et** dévermineur GDB.*
- ✓ *https://godbolt.org/ : La référence (car le premier…), pas forcément le meilleur*
- ✓ *https://www.programiz.com/cpp-programming/online-compiler/*
- ✓ *https://cpp.sh/*
- ✓ *…*

[1] TD : ½ groupe, apports de cours, mise en pratique guidée – TP : ½ groupe, travail autonome majoritaire.
[2] Le niveau se rapporte à la thématique principale. Il peut ici être entendu comme un indicateur de la difficulté du travail (N-Facile, A-Sans grande difficulté, M-Quelques points complexes, MA-Difficile)
[3] Les niveaux cible correspondent pour chaque compétence opérationnelle au niveau d'acquisition de la celle-ci à l'issue du travail dans son intégralité (en incluant les phases préparatoires et de synthèse).
[4] En plus d'un environnement de développement C++

Travail encadré en séance

Apports de cours

- ✓ void 😊

Activités

Activité 1 : Le fameux « Hello World ! »

- Recopier le code suivant dans un fichier texte qui sera sauvegardé avec « l'extension » .cpp

```
// Your First C++ Program

#include <iostream>

int main() {
    std::cout << "Hello World!" << std::endl ;
    return 0;
}
```

- Identifier la ligne de commande permettant de produire un exécutable à partir de ce fichier source (avec GCC ou votre compilateur). Identifier la ou les options utilisées.
- Imaginer ce que fait ce programme. Valider l'hypothèse en l'exécutant.
- Identifier précisément le rôle de chaque ligne de ce programme. Proposer éventuellement des alternatives et discuter de la pertinence d'utiliser l'une ou l'autre des possibilités.

Activité 2 : Découverte des fonctions en C++

- Coder et tester une fonction « DireBonjour », ne prenant aucun paramètre et ne retournant rien, dont le rôle est d'afficher sur la console le message « Bonjour ».
  - Mettre en place une architecture de projet C++ « standard »
- Réfléchir aux moyens de disposer de la possibilité d'enrichir le message en fournissant un nom à la fonction lorsque le besoin s'en fait sentir.
  - Quel type pour le nom ?
  - Tester la/les propositions
- Coder une fonction « CalculerSortieCapteur » qui prend en entrée 3 paramètres (double) représentant respectivement l'entrée du capteur, sa sensibilité et un éventuel offset (valeur par défaut = 0).
  - Tester le comportement. Identifier le moyen de « passer » une valeur « non string » au flux de sortie cout.
- Analyser et coder une fonction « Swap » qui échange les valeurs de deux entiers passés en paramètre.
  - Identifier la stratégie à employer ici quant au passage des paramètres.
  - Identifier un outil de la STL rendant l'opération plus « moderne »
- Proposer une solution permettant d'étendre cette fonction à d'autres types. Cette nouvelle s'appellera « templateSwap ». On demandera, de plus à cette fonction de retourner la taille (en octets) de l'objet.
  - Identifier une « contrainte » de l'utilisation des templates.
  - Quel est l'intérêt par rapport à la surcharge ?

Activité 3 : Travail autour des structures

Le travail destiné à découvrir les structures en C++ (et bien d'autres spécificités…) se base sur deux structures :

- ✓ Une structure *point*, composée de trois champs :
  - x : position en x du point (uint16_t)
  - y : position en y du point (uint16_t)
  - Couleur : couleur du point (couleur_t)
- ✓ Une structure *couleur_t*, composée de trois champs de type uint8_t:
  - Red : composante rouge
  - Green : composante verte
  - Bleu : composante bleue

Une fois ces structures déclarées, il s'agira de développer quelques fonctions mettant en œuvre des objets de type point et/ou couleur_t.

- ➢ Déclarer dans le « .hpp » les structures. Vérifier qu'il est possible de créer un objet de type point, de type couleur_t et d'accéder aux différents champs.
  - Mettre en évidence les différentes manières de définir et d'initialiser des objets de ces types.
  - On pourra, pour valider les hypothèses, s'appuyer sur une fonction « dump » qui a pour rôle d'envoyer vers la console les 5 champs d'un point.
    - Quel est le type du paramètre d'entrée de cette fonction ?
- ➢ Analyser et coder une fonction « Move » dont le rôle est de « déplacer » le point d'un deltaX et d'un DeltaY (int16_t).
  - Un mouvement doit être contenu dans la gamme [0 ; 65535].
  - La fonction retourne un booléen :
    - true si le déplacement a bien été effectué
    - false si le point a rencontré un « bord »
- ➢ Analyser, coder et tester une fonction « Fuzz » qui retourne un point construit à partir de deux points passés en paramètres. Les champs de ce nouveau point valent la moyenne (arrondie) des champs des deux points passés en paramètre.
  - Quel type retourner ? Pourquoi ? Faire plusieurs tests et essais.

Activité 4 – Types et Expressions constantes

- ➢ Si la manipulation n' a pas été faite auparavant, définir un objet de type const point appelé « Origine » (0,0,255,255,255).
  - Que se passe-t-il si l'on passe cet objet à la fonction « Move » ? Est-ce normal ?
- ➢ Dans le .hpp créer deux expressions constantes destinées à créer des couleurs « Black » et « White ».
  - Remplacer le « code couleur » de Origine par White. Expliquer ce qu'il se passe au moment de la compilation.

Synthèse

Faire le bilan de ce qui a été vu et abordé lors de ce TD. Revenir sur ce qui n'a pas été assez complètement compris.

Après la séance…

Faire des recherches complémentaires sur les expressions constantes. Le but est de bien comprendre l'intérêt de celles-ci et notamment d'aborder les notions de « compile-time » et « runtime ».