


Université de Sousse École pluridisciplinaire internationale Département informatique	 EPI <small>ÉCOLE PLURIDISCIPLINAIRE INTERNATIONALE</small>	A.U. : 2019/2020 Niveau : 2ème Année Préparatoire Module : POO C++ (TP) Enseignante : Maïssa HAMOUDA
---------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------

LANGAGE C++

Maïssa HAMOUDA

Plan

1. Structure d'un programme : Variables, Type de données (primitifs, tableaux, chaînes...), Opérateurs, Entrée-Sortie, Structures de contrôles...
2. Composition de classes
3. Héritage et polymorphisme
4. Les paquetages (package)
5. Les classes abstraites
6. Héritage multiple et les Interfaces
7. Les exceptions

28/01/2020

Maïssa HAMOUDA

2

Introduction à la programmation OO 1/2

• Notion d'objets

- La programmation orientée objet consiste à définir des objets logiciels et à les faire interagir entre eux.
- Concrètement, un objet est une structure de données (ses attributs = des variables) qui définit son état et une interface (ses méthodes = des fonctions) qui définit son comportement.
- Un objet est créé à partir d'un modèle appelé classe. Chaque objet créé à partir de cette classe est une instance de la classe en question.

• Notion de classe

- Une classe déclare des propriétés communes à un ensemble d'objets. Une classe représentera donc une catégorie d'objets. Elle apparaît comme un type à partir duquel il sera possible de créer des objets.

28/01/2020

Maïssa HAMOUDA

3

Introduction à la programmation OO 2/2

• Notion de visibilité

- Le C++ permet de préciser le type d'accès des membres (attributs et méthodes) d'un objet. Cette opération s'effectue au sein des classes de ces objets :
 - public : les membres publics peuvent être utilisés dans et par n'importe quelle partie du programme.
 - privé (private) : les membres privés d'une classe ne sont accessibles que par les objets de cette classe et non par ceux d'une autre classe.

• Notion d'encapsulation

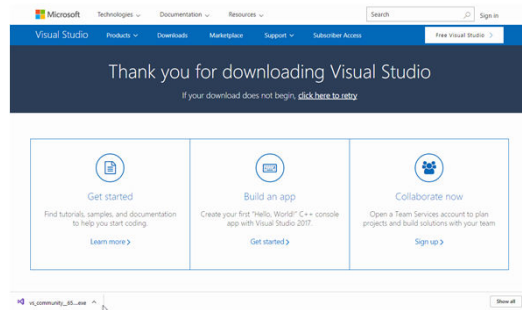
- L'encapsulation est l'idée de protéger les variables contenues dans un objet et de ne proposer que des méthodes pour les manipuler. En respectant ce principe, toutes les variables (attributs) d'une classe seront donc privées. L'objet est ainsi vu de l'extérieur comme une "boîte noire" possédant certaines propriétés et ayant un comportement spécifié.

28/01/2020

Maïssa HAMOUDA

4

Installation de Visual studio C++

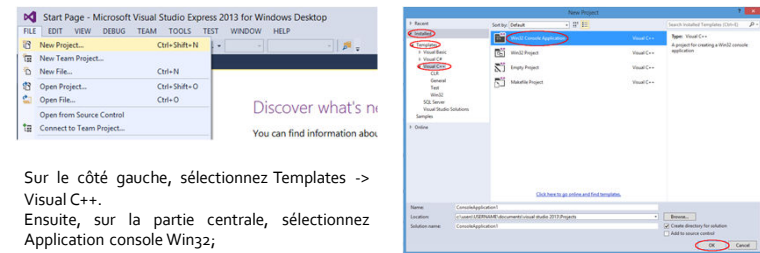


28/01/2020

Maïssa HAMOUDA

5

Création d'un projet 1/3



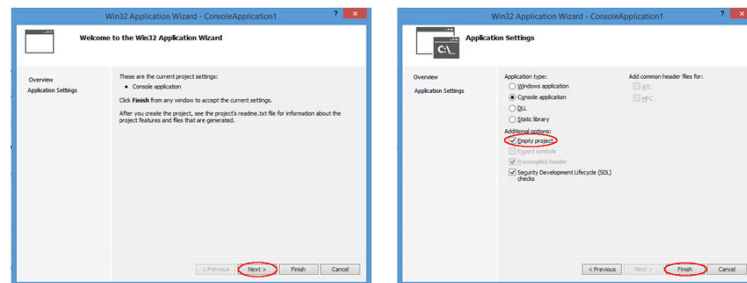
Sur le côté gauche, sélectionnez Templates -> Visual C++.
Ensuite, sur la partie centrale, sélectionnez Application console Win32;

28/01/2020

Maïssa HAMOUDA

6

Création d'un projet 2/3



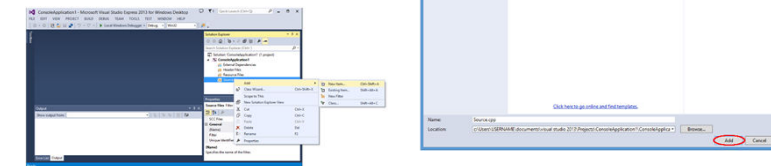
28/01/2020

Maïssa HAMOUDA

7

Création d'un projet 3/3

1. Sélectionnez projet Empty .
2. Maintenant, nous avons un projet vide, nous devons y ajouter un fichier.
3. Pour cela: dans l' Explorateur de solutions à droite, recherchez les fichiers source sous votre application.
4. Cliquez avec le bouton droit -> Ajouter -> Nouvel élément ...



Ici, ajoutez un nouveau fichier C++:

28/01/2020

Maïssa HAMOUDA

8

Premier programme en C++ (1/2)

- Écrivez-y ce qui suit:

```
// my first program in C++
#include <iostream>
int main()
{
    std::cout << "Hello World!";
}
```

- Ensuite, pour compiler et exécuter cette application, appuyez simplement sur Ctrl+F5.
- Vous pouvez modifier ce fichier autant que nécessaire et déclencher une nouvelle compilation et exécution à chaque fois que vous êtes prêt.

28/01/2020

Maïssa HAMOUDA

9

Premier programme en C++ (2/2)

Directive indiquant au préprocesseur d'inclure une section de code C++ standard, connue sous le nom d'iostream d'en-tête, qui permet d'effectuer des opérations d'entrée et de sortie standard.

Commentaire

```
// my first program in C++
#include <iostream>
int main()
{
    std::cout << "Hello World!";
}
```

Lance la déclaration d'une fonction. L'exécution de tous les programmes C++ commence par la main fonction, quel que soit son emplacement réel dans le code.

Instruction comportant 4 parties:

1. std::cout identifie le caractère standard sur l'écran d'ordinateur.
2. L'opérateur d'insertion (<<) indique que ce qui suit est inséré dans std::cout.
3. Une phrase entre guillemets ("Bonjour tout le monde!") est le contenu inséré dans la sortie.
4. « ; » indique la fin de l'instruction.

Maïssa HAMOUDA

10

Modification du programme

- Maintenant, ajoutons une déclaration supplémentaire à notre premier programme:

- std::cout << "Je suis un programmeur C++";

- Utilisation de l'espace de noms std

- using namespace std;

- Les deux façons d'accéder aux éléments de l'espace de noms std sont valides en C++ et produisent exactement le même comportement.

```
// my second program in C++
#include <iostream>
using namespace std;
int main ()
{
    cout << "Hello World! ";
    cout << "Je suis un programmeur C++";
}
```

28/01/2020

Maïssa HAMOUDA

11

Variables et types

- Chaque variable a besoin d'un nom qui l'identifie et le distingue des autres.
- Type de données :
 - Caractères**: ils peuvent représenter un seul caractère, tel que 'A' ou '\$'. Le type le plus élémentaire est char, qui est un caractère à un octet. D'autres types sont également fournis pour les caractères plus larges.
 - Entiers numériques**: ils peuvent stocker une valeur entière, telle que 700 1024. Ils existent en différentes tailles et peuvent être signés ou non, selon qu'ils prennent en charge des valeurs négatives ou non.
 - Type booléen**: le type booléen, connu en C++ sous le nom bool, ne peut représenter qu'un des deux états, true ou false.

```
// operating with variables
#include <iostream>
using namespace std;
int main ()
{
    // declaring variables:
    int a, b;
    int result;
    // process:
    a = 5;
    b = 2;
    a = a + 1;
    result = a - b;
    // print out the result:
    cout << result;
    // terminate the program:
    return 0;
}
```

28/01/2020

Maïssa HAMOUDA

12

Chaines de caractères

- Pour déclarer et utiliser des objets (variables) de ce type, le programme doit inclure l'en-tête où le type est défini dans la bibliothèque standard (en-tête <string>):

```
// my first string
#include <iostream>
#include <string>
using namespace std;

int main ()
{
    string mystring;
    mystring = "This is a string";
    cout << mystring;
    return 0;
}
```

28/01/2020

Maïssa HAMOUDA

13

Opérateurs

- Opérateur d'affectation (=)
- Opérateurs arithmétiques (+, -, *, /, %,)
- Affectation composée (+ =, - =, * =, / =, % =, >> =, << =, & =, ^ =, | =)
- Incrémenter et décrémenter (++ , --)
- Opérateurs relationnels et de comparaison (==, !=, >, <, >=, <=)
- Opérateurs logiques (!, &&, ||)
- Opérateur ternaire conditionnel (?)
- Opérateur virgule (,)
- Opérateurs au niveau du bit (&, |, ^, ~, <<, >>)
- Opérateur de transtypage de type explicite (exemple : i = (int) f;)
- Taille : x = sizeof (char);

28/01/2020

Maïssa HAMOUDA

14

Entrée / sortie de base 1/2

- La bibliothèque standard définit une poignée d'objets de flux qui peuvent être utilisés pour accéder à ce qui est considéré comme les sources et destinations standard des caractères par l'environnement dans lequel le programme s'exécute:

cin flux d'entrée standard
 cout flux de sortie standard
 cerr flux d'erreur standard (sortie)
 clog flux de journalisation (sortie) standard

Cette instruction oblige le programme à attendre l'entrée de cin, généralement avec le clavier.

Sortie standard (cout)

```
cout << "Hello"; // prints Hello
cout << Hello; // prints the content of variable Hello
```

Entrée standard (cin)

```
int age;
cin >> age;
```

28/01/2020

Maïssa HAMOUDA

15