**Exercices : Programmation structurée en C#**

**Exercice 0 : test généralités sur c#**

1. C# est un langage :

a. Déclaratif

b. Procédural

c. Evènementiel

2. C# est un langage :

a. Interprété

b. Précompilé puis compilé

c. Compilé

3. En C#, il y a :

a. Des variables

b. Des instructions

c. Des constantes

4. C# est un langage

a. Typé fortement

b. Non typé comme Python.

**Exercice 1 : débutons en c#**

Ecrire le programme ci-dessous avec Visual Studio : (*créer un nouveau projet, choisir un projet console (.Net), nommer votre 1ier programme ex1, entre les accolades du main écrire :)*

int a ;

string b,y ;

a = 1 ;

a=a+10 ;

b="bonjour " ;

y=" Hedi Lamar" ;

b=b+y ;

***1.*** ***Exécuter***

***2.*** ***Que remarquez-vous ? sur la fenêtre où vous avez écrit le code***

*Différentes instructions avec utilisation de couleurs différentes pour les types, les littéraux, les noms de variables*

***3.*** ***Que remarquez-vous ? sur la fenêtre noire***

Rien n’apparaît.

***4.*** ***Ajouter en dernière ligne*** : a = a + y;

***5.*** ***Que remarquez-vous ?***

Une erreur dans le code. (Une vaguelette rouge apparaît, une suggestion sous la forme d’ampoule)

***6.*** ***Expliquer.***

Visual Studio nous signale une erreur de compilation : une incompatibilité de type . C’est un éditeur de développement intégré

**Exercice 2 :**

**Qu’avez-vous appris grâce à ce programme ?**

Corriger nos erreurs

.  **Qu’avez-vous appris grâce à cet exercice ?**

Utiliser Visual Studio , Créer un nouveau projet, Nommer le projet et choisir le lieu où l’enregistrer, Application Console C#, Ecrire le code source sous Visual Studio, Exécuter le programme ,Comprendre nos erreurs de compilation

**Exercice 3 :**

Barrer les noms qui ne sont pas valides pour C#.

Ecrire s’il respecte la politique de nommage énoncée dans le cours :

nbBandesDessinees NB+lignes NbLignes @float flo 1metreCarre \_A123

**Exercice 4 :**

a) Avec Visual Studio, écrire :

int i,j,k;

k=i+j ;

Console.WriteLine(k);

Console.ReadLine();

***1.*** ***Que remarquez-vous ? sur la fenêtre où vous avez écrit le code***

Des erreurs de compilation

***2.*** ***Que remarquez-vous ? sur la fenêtre d’exécution***

Exécution impossible

**b)** **Jouer à l’ordinateur avec le programme suivant,**

Avec Visual Studio, écrire :

int i=3,j=56,k=5;

k=i+j ;

Console.WriteLine(k);

Console.ReadLine();

***1.*** ***Que remarquez-vous ? sur la fenêtre d’exécution***

1. Aucune erreur de compilation.
2. A l'exécution, il affiche la valeur de k

***Séquence algo :***

a) Traduire l’algorithme suivant, en c# :

**Variables** a,b,c : entiers // Déclaration de variables

res : chaine de caractères

**Début**

Afficher (« écrire a »)

Lire(a)

Afficher (« écrire b»)

Lire(b) // l'ordinateur lit sur le bus du clavier

c<-a\*b // c<- signifie c est affecté par

res<-"le résultat est "+c

Afficher (res)

**Fin**

**Exercice 5 : vérifier si les types de la déclaration de la variable et de la constante affectée sont compatibles**

Avec Visual Studio, vérifier si les types, les variables et les valeurs sont **compatibles** :

Double d=1.2 ;

float f1=1.2 ;

float f2 = 1.2 F ;

float f3 = (float) 1.2 ;

float f4 = 1.2e10f ;

short s=10 ;

double dd=1.2f ;

**Exercice 6 : tester les caractères de contrôle**

**a)** Avec Visual Studio, exécuter les lignes suivantes :

Console.WriteLine("toto");

Console.Write("tutu ");

Console.Write("rose ");

Console.ReadLine();

***1.*** ***Quelles différences, remarquez-vous entre Write et WriteLine ?***

***2.*** ***Ajouter juste après tutu*** \n , exécuter **puis** \r ? ***Que constatez-vous ?***

***3.*** ***A la place*** \n, ajouter ***juste après tutu*** \t ? ***Que constatez-vous ?***

**Exercice 7 : chercher la constante cachée et la declarer en tant que constante**

1. Ouvrir le fichier exemple7.cs avec Visual Studio.

2. Avant de l’exécuter, commenter chaque ligne de ce programme pour mieux le comprendre.

3. Jouer à la machine sur les lignes de ce programme.

4. Exécuter le avec Visual Studio

5. Quelle est la variable, qui pourrait être transformée en constante ?

6. Expliquer pourquoi vous avez fait ce choix. Quel en est l’intérêt ?

7. Modifier le programme pour prendre en compte cette remarque.

8. Exécuter le avec Visual Studio