ECOLE NORMALE SUPÉRIEURE DE
L'ENSEIGNEMENT TECHNIQUE
DE MOHAMMEDIA
UNIVERSITÉ HASSAN II DE
CASABLANCA

ENSET

المدرسة العليا لأساتذة التعليم التقني
المحمدية
جامعة الحسن الثاني بالدار البيضاء

Travaux Pratiques

COURS

Ce TP réalisé par :

ZRAIDI NAJWA
(GLSID1)

Ecriture de données sur la sortie standard

Exemple 1 :

- ✓ Ce programme permet d'afficher une chaine de caractère avec la fonction standards« cout » et l'opérateur « << » .

Exemple 1.cpp | Exemple 2.cpp

```cpp
#include<iostream>
int main()
{
    std :: cout<<" Langage C++ ";
}

/* ou
#include<iostream>
using namespace std;
int main()
{
    cout<<" Langage C++ ";
}
ce code donne la meme resultat de le code compiler qui permet d'afficher une chaine de caractere.
*/
```

```
C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP1\Cours\Ecriture de...
 Langage C++
--------------------------------
Process exited after 0.05506 seconds with return value 0
Appuyez sur une touche pour continuer...
```

Exemple 2 :

- ✓ Ce programme permet d'afficher une chaine de caractère avec la fonction standards« cout » et l'opérateur « << » et j'ai la possibilité de mettre les variables avec cette chaine de caractère .

Exemple 1.cpp | Exemple 2.cpp

```cpp
#include<iostream>
using namespace std;
int main()
{
    float Pi=3.14;
    cout<< "La valeur de Pi est : ";
    cout<< Pi ;
    // la meme chose de mettre  cout<< "La valeur de Pi est : " << Pi ;
    cout<<endl; // endl : reteur à la ligne ou \n comme
    cout<< "La valeur de Pi est : " << Pi ;
}
```

```
C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP1\Cours\Ecr...
La valeur de Pi est : 3.14
La valeur de Pi est : 3.14
--------------------------------
Process exited after 0.07843 seconds with return value 0
Appuyez sur une touche pour continuer...
```

Lecture de données sur l'entrée standard

Exemple :

- ✓ Ce programme permet d'afficher et de lire une chaine de caractère avec les fonctions standards« cout », « cin » , les opérateurs « << », « >> » et j'ai la possibilité de mettre les variables avec cette chaine de caractère .

Exemple.cpp

```cpp
#include<iostream>
using namespace std;
int main()
{
    int N;
    cout<< "Entrer un nombre entier : ";
    cin>> N;
    cout<< "Le carre du nombre entre est : "<<N*N;
}
```

```
C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP1\Cours\Lecture de donnÚes su...
Entrer un nombre entier : 4
Le carre du nombre entre est : 16
--------------------------------
Process exited after 2.739 seconds with return value 0
Appuyez sur une touche pour continuer...
```

Les conversions explicites

Exemple 1 :

Exemple 1.cpp | Exemple 2.cpp

```cpp
#include<iostream>
using namespace std;
int main()
{
  double d=3.145;
  int i;
  i=d; // ce type de conversation appellé "conversation implicite"
  cout << "La valeur de d : "<<i<<endl;
  i =(int) d;// ce type de conversation appellé "conversation explicite"
  cout << "La valeur de d : "<<i;

}
```

```
C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP1\Cours\Les conv...
La valeur de d : 3
La valeur de d : 3
--------------------------------
Process exited after 0.05993 seconds with return value 0
Appuyez sur une touche pour continuer...
```

Exemple 2 :

Exemple 1.cpp | Exemple 2.cpp

```cpp
#include<iostream>
#include<conio.h>
using namespace std;
int main()
{
   char c='m',d=25.5,e;
   int i=42,j;
   float r=678.9,s;
   j=c;
   cout<<j<<"\n"; // j prend la valeur de m à encodage assci car j est un entier donc j=109
   j=r;
   cout<<j<<"\n"; // j prend la valeur de r avec la conversion implicite alors j=678
   s=d;
   cout<<s<<"\n";
   /*s prend la valeur de i avec la conversion implicite d'ou s=25.5 mais !!!! d'après le compilateur
   fait la conversation impilicite de char à un entier donc il affiche s=25 */
   e=i;
   cout<<e<<"\n";// e prend la valeur de i et convertir  à encodage assci car i est un entier donc e=*
   getch();
}
```

```
C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP1\Cours\Les conve...
109
678
25
*

--------------------------------
Process exited after 1.914 seconds with return value 0
Appuyez sur une touche pour continuer...
```

Visibilité des variables

Exemple :

Exemple 1.cpp | Exemple.cpp

```cpp
#include<iostream>
using namespace std;
int i=11; // declaration d'un variable global
int main()
{
    int i=34;
    {
        int i=23; // declaration d'un variable local dans le bloc
        :: i= :: i+1;
    cout<< :: i <<" "<<i<<endl;
    }
  cout<< :: i <<" "<<i<<endl;

}
```

```
C:\Users\Najwa\Desktop\TP_CPP_ZRAIDI Najwa\TP1\Cours\Visibilite des variables\Exempl...
12 23
12 34

--------------------------------
Process exited after 0.06323 seconds with return value 0
Appuyez sur une touche pour continuer...
```

Valeur par défaut des paramètres

Exemple :

Exemple 1.cpp

```cpp
#include<iostream>
using namespace std;

void f1(int n=3){//par défaut le paramètre n vaut 3
    //traitement
}
void f2(int n,float x=2.35){
    //traitement
}
void f3(char c,int n=3,float x=2.35){
    //traitement
}
int main(){
    char a=0; int i=2; float r=5.6;
    f1(i);//l'arguemt n vaut 2, l'initialisation par défaut est ignorée
    f1();// L'argument n prend la valeur par défaut : 3
    f2(i,r); // les intialisations par defaut sont ignorées
    f2(i); //le second pramètre prend la valeur par défaut !!! f2(); interdit car Dans la fonction a des arguments
    f3(a,i,r); // les initialisations par defaut sont ignorées
    f3(a,i); //Le troisième paramètre prend la valeur par defaut
    f3(a); // le deuxième et la troisième paramètres prennent les valeurs par defaut

}
```

Surcharge des fonctions

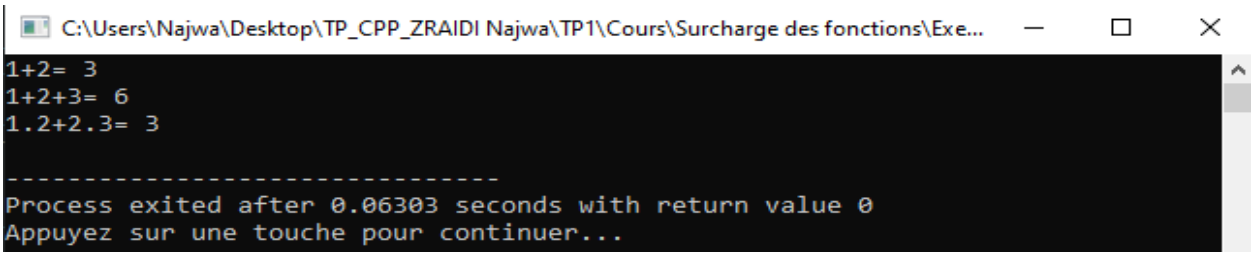
Exemple :

Exemples.cpp | Exercice.cpp

```cpp
#include<iostream>
using namespace std;

int somme(int n1,int n2){
    return n1+n2;
}
int somme(int n1,int n2,int n3){
    return n1+n2+n3;
}
int somme(double n1,double n2){
    return n1+n2;
}
int main(){

    cout<<"1+2= "<<somme(1,2)<<endl; //Affichage la somme des deux nombres entiers
    cout<<"1+2+3= "<<somme(1,2,3)<<endl; //Affichage la somme des trois nombres entiers
        cout<<"1.2+2.3= "<<somme(1.2,2.3)<<endl; //Affichage la somme des deux nombres
}
```



Exercice :

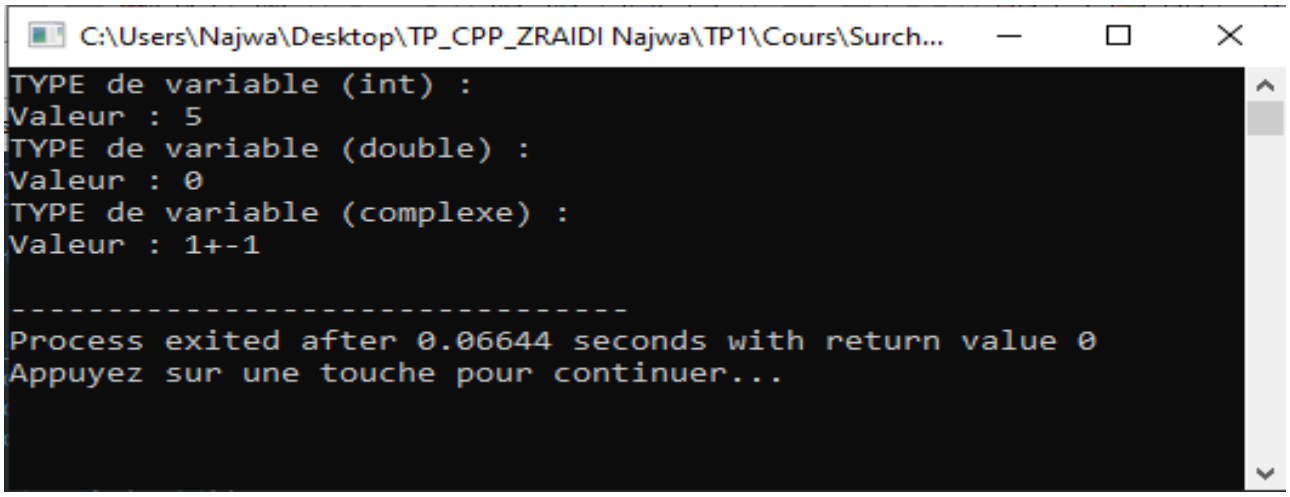
Exemples.cpp | Exercice.cpp

```cpp
#include<iostream>
using namespace std;

struct Complexe
{
    double reel,im;// reel est la partie réel  et im est la partie imaginaire d'un nombre complexe
};
void affiche(int); // declaration de la fonction affiche
void affiche(double); // declaration de la fonction affiche
void affiche(Complexe);// declaration de la fonction affiche

int main(void){
    int a=5;
    double d=0.0;
    Complexe c={1.0,-1.0};
    affiche(a);//Appel la fonction (1)
    affiche(d);//Appel la fonction (2)
    affiche(c);//Appel la fonction (3)
}
void affiche(int i){
    cout<< "TYPE de variable (int) : "<<endl;
    cout<<"Valeur : "<<i<<endl;
}
void affiche(double d){
    cout<< "TYPE de variable (double) : "<<endl;
    cout<<"Valeur : "<<d<<endl;
}
void affiche(Complexe c){
    cout<< "TYPE de variable (complexe) : "<<endl;
    cout<<"Valeur : "<<c.reel<<"+"<<c.im<<endl;

}
```



Allocation mémoire

Exemple 1 :

Exemple1.cpp | Exemples.cpp

```cpp
#include<iostream>
using namespace std;
struct Complexe
{
    double reel,im;// reel est la partie réel  et im est la partie imaginaire d'un nombre complexe
};
int main(){
    Complexe *Z;
    Z=new Complexe[50];
    char *pc=new char[100];
    delete pc; //desalloue la zone de 100 caratères
    delete Z; //ne libère que le premier élément
    delete []Z;//ou delete [50]Z;
}
```

Exemple 2 :

Exemple1.cpp | Exemples.cpp

```cpp
#include<iostream>
using namespace std;

int main(){
    int *pi;//déclaration du pointeur
    pi=new int;//allcation de la mémoire
    /* ou aurait pu écrire :
    int *pi = new int; // Allocation de memoire en c est : pi=(int*)malloc(sizeof(int));
    */
    int *ptr1,*ptr2,*ptr3; // declaration des poitures
    //Allocation dynamique d'un entier
    ptr1=new int;
    //Allocation d'un tableau de 10 entiers
    ptr2=new int[10];
    //Allocation d'un entier avec intialisation
    ptr3=new int(10);
    // structure de date qui regroupe jour,mois et année
    struct date {int jour,mois,an;};
    date *ptr4,*ptr5,*ptr6,d={25,4,1952};
    //Allocation dynamique d'une structure
    ptr4=new date;
    //Allocation dynamique d'un tableau de structure
    ptr5=new date[10];
    //Allocation dynamique d'une structure avec intialisation
    ptr6=new date(d);
}
```