2022

ECOLE NORMALE SUPÉRIEURE DE L'ENSEIGNEMENT TECHNIQUE

DE MOHAMMEDIA

- UNIVERSITÉ HASSAN II DE CASABLANCA

TP1 : programmation orientée objet en c++

ETUDIANT DE FILIER GLSID 1

ENSET

- ENCADRÉ PAR:
  M.K.MANSOURI
- REALISER PAR :
  ZAKARIA EL MOURTAZAK

EXERCICE 1

LE CODE

ex1.cpp > main(void)

```cpp
#include <iostream>
using namespace std;
int main(void)
{
    int n;
    float x;
    cout << "donnez un entier et un flottant\n";
    cin >> n >> x;
    cout << "le produit de " << n
        << " par " << x
        << "\n est " << n * x;
    return 0;
}
```

L'EXECUTION

```
donnez un entier et un flottant
2 2.4
le produit de 2 par 2.4
 est 4.8
```

EXERCICE 2

LE CODE

```
2  #include <iostream>
3  #include <conio.h>
4  using namespace std;
5  main()
6  {
7      int i, n = 25, *p;
8      char *ch = "On est à l'IGA !";
9      float x = 25.359;
10     cout << "BONJOUR\n";
11     cout << ch << "\n";
12     cout << "BONJOUR\n" << ch << "\n";
13     cout << "n= " << n
14          << " x= " << x
15          << " p = " << p << "\n ";
16     getch();
17 }
```

L'EXECUTION

```
BONJOUR
On est á l'IGA !
BONJOUR
On est á l'IGA !
n= 25 x= 25.359 p = 0x1
```

EXERCICE 3

QUESTION N1

LE CODE

```cpp
#include <iostream>
#include <conio.h>
using namespace std;
main()
{
    int n;
    char tc[30], c;
    float x;
    cout << "Saisir un entier:";
    cin >> n;
    cout << "Saisir un réel:";
    cin >> x;
    cout << "Saisir une phrase:";
    cin >> tc;
    cout << "Saisir une lettre:";
    cin >> c;
    cout << "Affichage : " << n << " " << x << " " << tc << " " << c << "\n"
    getch();
}
```

L'EXECUTION SIMPLE

```
Saisir un entier:2
Saisir un r@el:2.3
Saisir une phrase:zakaria
Saisir une lettre:z
Affichage : 2 2.3 zakaria z
```

L'EXECUTION AVEC DES ERREURS DE SAISIE

```
Saisir un entier:3
Saisir un r├®el:zakaria
Saisir une phrase:Saisir une lettre:Affichage : 3 0 ▸
PS C:\Users\user\Documents\tpCpp>
```

```
Saisir un entier:3.4
Saisir un r├®el:Saisir une phrase:hk
Saisir une lettre:k
Affichage : 3 0.4 hk k
```

```
Saisir un entier:2
Saisir un r├®el:2.3
Saisir une phrase:zakaria elmourtazak
Saisir une lettre:Affichage : 2 2.3 zakaria e
```

```
Saisir un entier:2
Saisir un r├®el:2.3
Saisir une phrase:zakaria
Saisir une lettre:zakaria
Affichage : 2 2.3 zakaria z
```

QUESTION N2

On en déduit premièrement que le programme peut bloquer la saisie si on fait des erreurs de saisie on prend le premier exemple dans le cas où on donne une chaine de caractère au lieu d'un nombre

Deuxièmement dons le cas d'entrée d'un flottant au lieu d'un entier où le compilateur sépare la valeur en deux parties, la part d'entière qu'il a donnée à la variable entière et la part flottant a la variable flottante

Troisièmement, nous trouvons l'erreur d'espace dans la chaîne dans laquelle le compilateur comprend l'espace comme la fin de la chaîne

Dernièrement, nous trouvons que si nous donnons une chaîne au lieu d'une lettre, le compilateur prend juste la première lettre de la chaîne

EXERCICE 4

QUESTION N1

LE CODE

```
5   float puissance(float val, int nb)
6   {
7       float P = 1;
8       for (int i = 1; i <= nb; i++)
9           P = P * val;
10      return P;
11  }
```

QUESTION N2

LE CODE

```cpp
#include <iostream>
#include <conio.h>
using namespace std;
float puissance(float val, int nb)
{
    float P = 1;
    for (int i = 1; i <= nb; i++)
        P = P * val;
    return P;
}

main()
{
    char c = 5;
    int i = 10,j = 6;
    float r = 5.246, r1, r2, r3, r4, r5;
    r1 = puissance(r,j);
    r2 = puissance(r, c);
    r3 = puissance (j, i);
    r4 = puissance (j,r);
    r5 = puissance(0, 4);
    cout << "r1 = " << r1 << "\n";
    cout << "r2 = " << r2 << "\n";
    cout << "r3 = " << r3 << "\n";
    cout << "24 = " << r4 << "\n";
    cout << "r5 = " << r5 << "\n";
    getch();
}
```

L'EXECUTION

```
r1 = 20843.4
r2 = 3973.21
r3 = 6.04662e+07
24 = 7776
r5 = 0
```

QUESTION N3

On en déduit pareil on permute les paramètres de la fonction la fonction s'exécute bien sans problème

QUESTION N4

LE CODE

```
float puissance(float val, int nb = 4 )
{
    float P = 1;
    for (int i = 1; i <= nb; i++)
        P = P * val;
    return P;
}
```

L'EXECUTION

```
r1 = 20843.4
r2 = 3973.21
r3 = 6.04662e+07
24 = 7776
r5 = 0
r6 = 757.379
```

EXERCICE 5

QUESTION N1

LE CODE

```cpp
#include <iostream>
#include <conio.h>
using namespace std;

void test(int n = 0, float x = 2.5)
{
    cout << "Fontion N1 : ";
    cout << "n = " << n << " x = " << x << "\n";
}

void test(float x = 4.1, int n = 2)
{
    cout << "Fontion N2 : ";
    cout << "n = " << n << " x = " << x << "\n";
}
main()

{
    int i = 5;
    float r = 3.2;

    test(i, r);
    test(r, i);
    test(i);
    test(r);
}
```

L'EXECUTION

```
Fontion N1 : n = 5 x = 3.2
Fontion N2 : n = 5 x = 3.2
Fontion N1 : n = 5 x = 2.5
Fontion N2 : n = 2 x = 3.2
```

QUESTION N2

En déduire que si on ne donne pas tous les paramètres, la fonction prendra les paramètres par défaut

EXERCICE 6

QUESTION N1

LE CODE

```
#include<iostream>
#include <conio.h>
using namespace std;

void essai(float x, char c, int n = 0)
{
    cout << "Fontion N1: x = " << x << " c = " << c << " n = " <<n << "\n
}


void essai(float x, int n)
{
    cout << "Fontion N2 : x = " << x << " n = " <<n << "\n";

}

main()
{
    char  l = 'z';
    int   u = 4  ;
    float y = 2.0;

    essai(y, l, u);
    essai(y, l);
    essai(y, u);
    essai(u, u);
    essai(u, l);
    essai(y, y, u);

    getch();
```

L'EXECUTION

```
Fontion N1: x = 2 c = z n = 4

Fontion N1: x = 2 c = z n = 0

Fontion N2 : x = 2 n = 4
Fontion N2 : x = 4 n = 4
Fontion N1: x = 4 c = z n = 0

Fontion N1: x = 2 c = ☻ n = 4
```

QUESTION N2

En déduire que dans le cas où les deux fonctions ont le même nom le compilateur décide de la fonction qui sera exécutée à partir du nombre et du type de paramètre

EXERCICE 7

QUESTION N1

LE CODE

```
void affiche (float x, int n = 0)
{
    if (x == 0)
    cout << "0^0 = 0\n" ;
    else
    cout << x << "^" << n << " = " << pow(x,n)<<"\n";
}
```

QUESTION N2

LE CODE

```
void affiche (int n,float x=0)
{
    if (n == 0)
    cout << "0^0 = 0\n" ;
    else
    cout << x << "^" << n << " = " << pow(x,n)<<"\n";
}
```

QUESTION N3

LE CODE

```
main ()
{
    int nb = 3;
    float val = 3.3;

    affiche (val);
    affiche (nb);
    affiche (val,nb);
    affiche (val,nb);

    getch();
}
```

L'EXECUTION

```
3.3^0 = 1
0^3 = 0
3.3^3 = 35.937
3.3^3 = 35.937
```

EXERCICE 8

QUESTION N1

LE CODE 1

```
#include<iostream>
#include <conio.h>
using namespace std;

void echange (int a, int b)
{
    int tampon;
    tampon = b;
    b = a   ;
    a   = tampon ;
    cout << "pendant l'echange : a = "<< a << "  b = " << b <<"\n";
}

main()
{
    int u = 5, v = 3;

    cout << "avant l'echange   : u = "<< u << "  v = " << v <<"\n";
    echange (u, v);
    cout << "apres l'echange   : u = "<< u << "  v = " << v <<"\n\n";

    getch();
}
```

L'EXECUTION 1

```
avant l'echange   : u = 5  v = 3

pendant l'echange : a = 3  b = 5
apres l'echange   : u = 3  v = 5
```

LE CODE 2

```cpp
#include<iostream>
#include <conio.h>
using namespace std;

void echange (int *a, int *b)
{
    int tampon;
    tampon  = *b;
    *b   = *a   ;
    *a   = tampon ;


    cout << "pendant l'echange : a = "<< *a << "  b = " << *b <<"\n";
}

main()
{
    int u = 5, v = 3;
    cout << "avant l'echange   : u = "<< u << "  v = " << v <<"\n\n";
    echange (&u, &v);
    cout << "apres l'echange   : u = "<< u << "  v = " << v <<"\n\n";

    getch();
}
```

L'EXECUTION 2

```
avant l'echange   : u = 5  v = 3

pendant l'echange : a = 5  b = 3
apres l'echange   : u = 5  v = 3
```

LE CODE 3

```cpp
#include <iostream>
#include <conio.h>
using namespace std;

void echange(int &a, int &b)
{
    int tampon;
    tampon = b;
    b = a;
    a = tampon;

    cout << "pendant l'echange : a = " << a << "  b = " << b << "\n";
}

main()
{
    int u = 5, v = 3;
    cout << "avant l'echange   : u = " << u << "  v = " << v << "\n\n";
    echange(u, v);
    cout << "apres l'echange   : u = " << u << "  v = " << v << "\n\n";

    getch();
}
```

L'EXECUTION 3

```
avant l'echange   : u = 5  v = 3

pendant l'echange : a = 3  b = 5
apres l'echange   : u = 3  v = 5
```

QUESTION N2

On constat que dans la première case les deux variables ne changent pas par contre les deux dernières qui changent près l'exécution de la fonction échange

QUESTION N3

On Conclut qu'avec le passage par adresse ou par référence on peut changer le contenu de variable par contre le passage par valeur le contenu de la variable ne change pas

EXERCICE 9

QUESTION N1

LE CODE

```
void affiche (float x, int n = 0)
{
    if (x == 0)
    cout << "0^0 = 0\n" ;
    else
    cout << x << "^" << n << " = " << pow(x,n)<<"\n";
}
```

QUESTION N2

LE CODE

```
void affiche (int n,float x=0)
{
    if (n == 0)
    cout << "0^0 = 0\n" ;
    else
    cout << x << "^" << n << " = " << pow(x,n)<<"\n";
}
```

QUESTION N3

LE CODE

```
main ()
{
    int nb = 3;
    float val = 3.3;

    affiche (val);
    affiche (nb);
    affiche (val,nb);
    affiche (val,nb);

    getch();
}
```

L'EXECUTION

```
3.3^0 = 1
0^3 = 0
3.3^3 = 35.937
3.3^3 = 35.937
```

EXERCICE 8

QUESTION N1

LE CODE / PAR REFERENCE

```
#include <iostream>
#include <conio.h>
using namespace std;

struct essai
{
    int n;
    float x;
};

void remis_a_zero(essai &e)
{
    e.n = 0;
    e.x = 0;
}

main()
{
    essai es;
    es.n = 10;
    es.x = 10.2;
    remis_a_zero(es);
    cout << "es.n = " << es.n
         << "  es.x = " << es.x << "\n\n";

    getch();
}
```

LE CODE / PAR ADRESSE

```
#include<iostream>
#include <conio.h>
using namespace std;

struct essai
{
    int n;
    float x;
};

void remis_a_zero(essai *e)
{
    e->n = 0;
    e->x = 0;
}

main()
{
    essai es;
    es.n = 5;
    es.x = 3.3; .


    remis_a_zero(&es);
    cout << "es.n = " << es.n << "  es.x = " << es.x<<"\n";

    getch();
```

L'EXECUTION DE CODE

```
  }
es.n = 0  es.x = 0
```

```cpp
#include<iostream>
#include <conio.h>
using namespace std;

struct essai
{
    int n;
    float x;
};

void remis_a_zero(essai *e)
{
    e->n = 0;
    e->x = 0;
}

main()
{
    essai es;
    es.n = 5;
    es.x = 3.3;


    remis_a_zero(&es);
    cout << "es.n = " << es.n << "  es.x = " << es.x<<"\n";

    getch();
```