

## FICHE RESUME SUR LES FONCTIONS EN C++

## EXEMPLE ELEMENTAIRE

```

/*****
Déclaration d'une fonction
OBLIGATOIRE SI VOUS APPELEZ UNE FONCTION
DEFINIE EN DESSOUS DE L'APPEL
*****/

```

```
int addition(int, int);
```

Le résultat est entier.

Ecrire **void** si la fonction n'a pas de résultat.

- Ne pas oublier le `;` à la fin de la ligne
- Les noms des paramètres sont facultatifs

```

/*****
Définition de la fonction
*****/

```

```
int addition(int a, int b)
{
    return a + b;
}
```

**a et b** sont des paramètres

**return** termine la fonction et renvoie le résultat à la fonction appelante.

```

/*****
Fonction principale
*****/

```

```
int main() // Fonction principale
{
```

```
    int x = 2;
```

```
    //===== APPEL DE LA FONCTION =====
```

```
    int somme = addition(x, 2);
```

```
    _getch(); // attente d'appui sur une touche
    return 0; // fin du programme. Le code 0 est envoyé
}
```

Ici, le passage de paramètres se fait **par valeurs**.  
x est copié dans a  
2 est copié dans b

## Le passage par référence

```

/*****
Permutation de 2 nombres
*****/

```

```
void permut(double &x, double &y)
{
    double tmp = x;
    x = y;
    y = tmp;
}
```

Le **&** indique un passage de paramètre **par référence**.  
**x** est un alias de **val1** (un autre nom) et **y** un alias de **val2**.  
Modifier **x** revient à modifier **val1**.  
Modifier **y** revient à modifier **val2**

```

/*****
Fonction principale
*****/

```

```
int main() // Fonction principale
{
```

```
    double val1 = 12.5, val2 = 7.4;
```

```
    permut(val1, val2);
    cout << "Après permutation val1=" << val1
         << " val2=" << val2 << endl;
```

```
    _getch(); // attente d'appui sur une touche
    return 0; // fin du programme. Le code 0 est envoyé
}
```

Après permutation val1=7.4 val2=12.5

## Les arguments par défaut

```

/*****
Calcul somme TTC
*****/

double ttc(double prixHT, double tva = 20)
{
    return prixHT + prixHT*tva / 100.0;
}

/*****
Fonction principale
*****/
int main()    // Fonction principale
{
    double ht , tva;
    char choix;

    cout << "Saisir le prix ht ";
    cin >> ht;
    cout << "Veux-tu une TVA standard (o/n) ? ";
    cin >> choix;

    switch(choix)
    {
        case 'o':
        case 'O':
            cout << "PRIX TTC= " << ttc(ht);
            break;

        case 'n':
        case 'N':
            cout << "Quel taux de tva veux-tu ? ";
            cin >> tva;
            cout << "PRIX TTC= " << ttc(ht,tva);
            break;
    }
}

```

Si le 2<sup>ème</sup> argument (tva) n'est pas fourni lors de l'appel de la fonction, il vaudra 20

Le taux de TVA n'a pas été fourni. La fonction choisira donc 20 %

Le taux de TVA a été fourni. La fonction choisira donc ce qu'on lui donne.

Ref : FicheResumeFonctions.docx mis à jour

BERT

## La surcharge (surdéfinition) des fonctions

```

/**** Aire d'un triangle équilatéral. Renvoi de l'aire ou -1 si erreur ****/
double calculAireTriangle(double a, double b, double c)
{
    double p; // le demi périmètre
    double tmp;

    p = (a + b + c) / 2;
    tmp = p*(p - a)*(p - b)*(p - c);
    if (tmp < 0) return -1;
    else return (sqrt(tmp));
}

/***** Aire d'un triangle isocèle *****/
double calculAireTriangle(double a, double b)
{
    double res;
    res= calculAireTriangle(a, b, b);
    return res;
}

/***** Aire d'un triangle équilatéral *****/
double calculAireTriangle(double a)
{
    return calculAireTriangle(a, a, a);
}

/***** Fonction principale *****/
int main()    // Fonction principale
{
    SetConsoleOutputCP(1252);
    double x, y, z , res;

    res = calculAireTriangle(2);
    if (res != -1)
        cout << "Aire du triangle equilateral de côtés 2 = " << res << endl;
    else cout << "Impossible de construire un triangle de côtés 2,2,2 " << endl;

    res = calculAireTriangle(2,7);
    if (res != -1)
        cout << "Aire du triangle isocèle de côtés 2,7,7 = " << res << endl;
    else cout << "Impossible de construire un triangle de côtés 2,7,7 " << endl;

    res = calculAireTriangle(2, 3 , 4);
    if (res != -1)
        cout << "Aire du triangle de côtés 2,3,4 = " << res << endl;
    else cout << "Impossible de construire un triangle de côtés 2,3,4 " << endl;

    res = calculAireTriangle(2, 7, 45);
    if (res != -1)
        cout << "Aire du triangle de côtés 2,7,45 = " << res << endl;
    else cout << "Impossible de construire un triangle de côtés 2,7,45 " << endl;

    _getch(); // attention
    return 0; // fin du programme
}

```

Ici, 3 fonctions portent le même nom, mais elles se distinguent par leur nombre de paramètres (elles pourraient aussi se distinguer par leurs types de paramètres)

C'est la fonction avec 3 arguments (paramètres) qui est appelée ici

Aire du triangle equilateral de côtés 2 = 1.73205  
 Aire du triangle isocèle de côtés 2,7,7 = 6.9282  
 Aire du triangle de côtés 2,3,4 = 2.90474  
 Impossible de construire un triangle de côtés 2,7,45