Travaux Dirigés de C++ n°2

Cours de programmation

—Licence d'informatique—

Surcharge, const, inline

Dans ce TD nous voyons la surcharge des fonctions, les constantes, et les fonctions inline.

▸ **Exercice 1.** *Écrire une macro* `MAX(a,b)` *qui calcule le maximum de* `a` *et* `b`*. Écrire une fonction* `int max(int a, int b)` *déclarée* `inline`*. Qu'affiche alors le programme suivant ?*

```
int main()
{ int a1=3, b1=4, c1;
  int a2=3, b2=4, c2;
  c1=MAX(a1++,b1++);
  c2=max(a2,b2);
  cout << "c1=" << c1 << endl;
  cout << "c2=" << c2 << endl;
  return 0;
}
```

▸ **Exercice 2.** *Que pensez vous des instructions de déclaration, d'initialisation et d'affectation suivantes et quelles sont leurs effets ?*

Avec des variables simples :

```
/* Initialisations */
int i = 12;
const int ic = 13;
const int jc = i;

/* Affectations */
ic = i;
jc = 13;
```

Avec des variables pointeurs :

```
/* Initialisations */
int* pi = &i;
int* pi2 = &ic;
```

```
int* const pci = &i;
int* const pci2 = &ic;
int* pci3 = &ic;

const int* const pcic = &ic;
const int* const pcic2 = &i;
const int* pcic3 = &i;

/* Affectations */
int j = 20;

pi = &j;
*pi = j;

pci = &j;
pcic = &j;
pcic3 = &ic;
```

Avec des variables références :

```
/* Initialisations */
int& ri = i;
int& ri = ic;
int& ri2 = 3;

int& const rci = i;

const int& ric = i;
const int& ric3 = ic;

/* Affectations */
ri = j;
ric = j;
ric = ic;
```

▸ **Exercice 3.** *Soient les deux fonctions déclarées de la façon suivante :*

```
int    min(int    a,int    b);
double min(double a,double b);
```

Quelles sont les fonctions appelées par :

```
min(3,4);
min(3.0,4);
min(3,4.0);
min(3.0,4.0);
```

On supprime la fonction de prototype `int min(int a,int b);`*. Que se passe t-il?*

▸ **Exercice 4.** *Soient les fonctions de profils suivants:*

```
void f(int);           // Profil 1
void f(const int);     // Profil 2
void f(int &);         // Profil 3
void f(const int &);   // Profil 4
```

Ces différentes déclarations sont-elles compatibles ? Si non, expliquez pourquoi. Lesquelles sont compatibles entre elles ? On supprime la seconde déclaration et on considère le code suivant:

```
int i=1;
const int ic=2;
int &r=i;
const int &rc=ic;

f(i);                 // Appel 1
f(ic);                // Appel 2
f(r);                 // Appel 3
f(rc);                // Appel 4
```

On supprime le second profil. Quels sont les appels de fonction ambigus ? Expliquez.

▸ **Exercice 5.** *Ecrire une fonction* `indexOf` *qui recherche l'indice de la position d'un caractère donné, dans une chaîne de caractères donnée, à partir d'une position donnée.*

Discuter de la possibilité d'utiliser des arguments par défaut pour les paramètres.

▸ **Exercice 6.** *Ecrire un petit programme en C++ qui utilise une fonction d'un module dont les sources sont écrits en C.*