Portée: variables locales/globales (1)

Les blocs ont une grande autonomie en C++.
Ils peuvent contenir leurs propres déclarations (et initialisations) de variables:

- les variables déclarées à l'intérieur d'un bloc seront appelées ***variables locales*** au bloc; elles ne sont accessibles (référençables) qu'à l'intérieur de ce bloc;

- les variables déclarées en dehors de tout bloc seront appelées ***variables globales*** au programme; elles sont accessibles dans l'ensemble du programme.

La portion de code dans laquelle une variable est utilisable est appelée la ***portée*** (ou *visibilité*[4]) de la variable, et s'étend usuellement de la déclaration de la variable jusqu'à la fin du bloc dans lequel est faite la déclaration

4. La visibilité effective d'une variable sera conditionnée par des règles de masquage qui seront vues plus loin dans le cours.

Portée: variables locales/globales (2)

```
// déclarations globales
int z(0);
...
void main()
{
    // déclaration locales
    int x,y;
    ...
    {
        // déclaration locales
        int y;
        ... x ...
        ... y ...
        ... z ...
    }
    ... y ...
    ... z ...
}
```

portée globale

portée locale fonction *main*

portée locale bloc interne

```
// début du programme
int z(0);
...
void main()
{
    // bloc niveau 1
    int x,y;
    ...
    {
        // niveau 2
        int y;
        ... x ...
        ... y ...
        ... z ...
    }
    ... y ...
    ... z ...
}
```

Portée: variables locales/globales (3)

En particulier, la déclaration d'une variable **à l'intérieur d'une boucle** est une déclaration **locale** au bloc associé à la structure de boucle.

Ainsi dans la structure de boucle

```
while (<condition>)
{
    int i(0);
    ...
}
```

la variable `i` est **locale** à la boucle

Et de même dans l'itération

```
for (int i(0); i<10; ++i)
{
    int j(0);
    cout << i << j << endl;
}
```

≡

```
{
    int i(0);
    while (i<10)
    {
        int j(0);
        cout << i << j << endl;
        ++i;
    }
}
```

Portée: variables locales/globales (4)

Exemple: le programme

```
const int MAX(5);
void main()
{
   int i(120);
   for (int i(1); i<MAX; ++i)
   {
      cout << i << endl;
   }
   cout << i << endl;
}
```

donne en sortie:

```
1
2
3
4
120
```

Portée: variables locales/globales (5)

Autre exemple: le programme

```
const int MAX(5);
void main()
{
   int i(120);
   int j(0);
   while (j<MAX)
   {
      int i(0);
      ++j;
      cout << j << ',' << i << endl;
      i=j;
   }
   cout << j << ',' << i << endl;
}
```

donne en sortie:

```
1,0
2,0
3,0
4,0
5,0
5,120
```