The C++ for ROOT cheat sheet

- The instruction blocks are enclosed by `{}`
- Lines can have any length, start and end anywhere
- Each instruction/line ends with `;`
- Upper case and lower case letters are distinguished:

`TheSame` ≠ **`theSame`** ≠ **`thesame`** ≠ **`Thesame`** ≠ **`THESAME`**

- All variables must be declared, but not necessarily at the beginning of the code block/program
- We can declare and initialise variables at the same time:

```
double MonGenou = 8.5;
```

The C++ for ROOT cheat sheet

- Variables have different types:
  - simple :

	`int`	`double`	`char`	`float`	`short int`
(f77)	*integer*4*	*real*8*	*character*	*real*4*	*integer*2*

  - complex:
    - association of several variables (*structure*)

```
struct maison{int colour; float number_of_floors;
float length; float width;}
```

    - structure with functions for manipulating the data variables (*class*)

```
class house{int colour; float number_of_floors;
 float length; float width;
  SetColour();GetColour();GetArea();}
```

  - arrays:

```
int h[10];double matrix[3][5];
house street[20];
```

The C++ for ROOT cheat sheet

• Loops

	FORTRAN Equivalent
`for(int i=0;i<10;i++) {}`	`do i=0,9 ... enddo`
`while (i != 10) {}`	`do while(i.ne.10) ... enddo`
`do {} while (k<=300)`	

• Logic

`==`	`.eq.`	`<`	`.lt.`	`||`	`.or.`
`!=`	`.ne.`	`<=`	`.le.`	`&&`	`.and.`
`!`	`.not.`	`>`	`.gt.`	0	`.FALSE.`
		`>=`	`.ge.`	≠0	`.TRUE.`

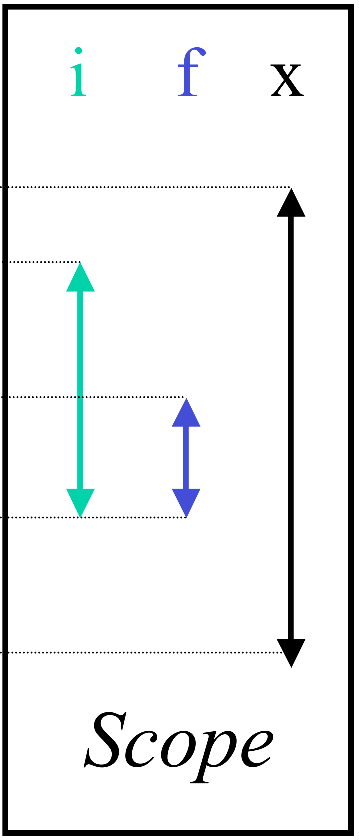
• If-Else

`if(i<10) {} else {}`	`if(i.lt.10) then ... else ... endif`

The C++ for ROOT cheat sheet

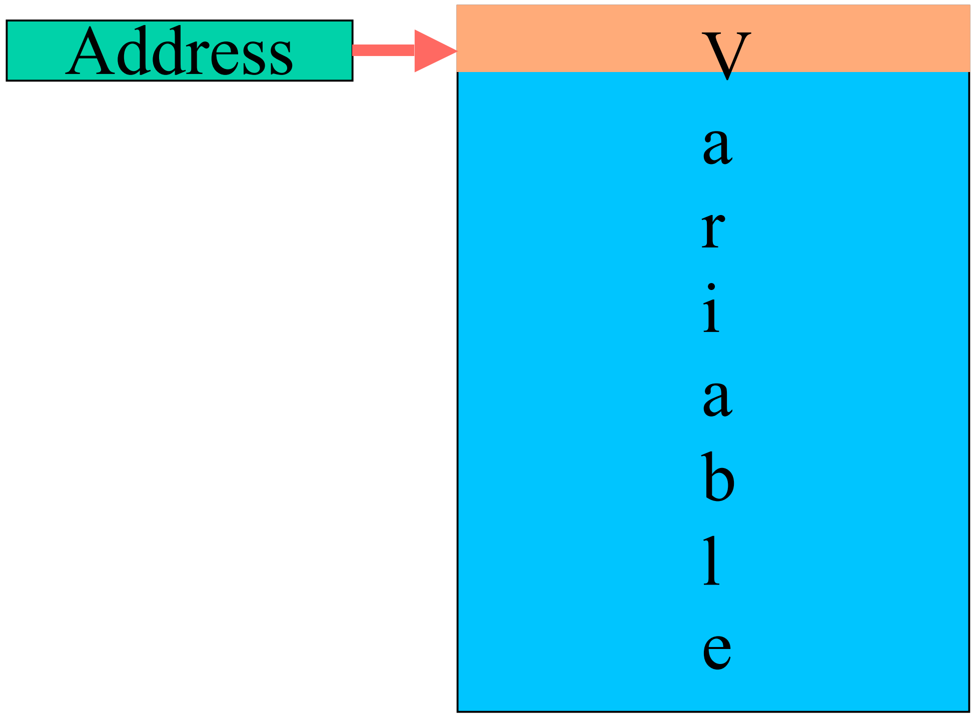
- Variables only exist in the block where they are declared (scope)

```
{
double x=3;
for(int i=0;i <10;i++)
 {
 double f=pow(x,i/2.);
 cout << x << "**" << i << "=" << f <<  endl;
 }
cout <<  "it' over!" << endl;
}
```



The C++ for ROOT cheat sheet

- Access to variables can be direct or via a *pointer*



```
Address   = &Variable

Variable = *Address
```

```
house SweetHome();
SweetHome.GetColour();
SweetHome.length;

house* addSH =&SweetHome;
addSH->GetColour();
addSH->length;
```

The C++ for ROOT cheat sheet

- Passing arguments to a function

```
void toto1(double a)
    {
    a=3;
    }
void toto2(double *a)
    {
    (*a)=15;
    }
void test_toto(void)
 {
 double x=8;
 toto1(x);
 cout << "X=" << x << endl;
 toto2(&x);
 cout << "X=" << x << endl;
 }
```

When function is called, argument is copied in **`a`** which is local to **`toto1`**.

When function is called, the address of the argument is copied in **`a`**.

`x` is not modified

`x` is modified

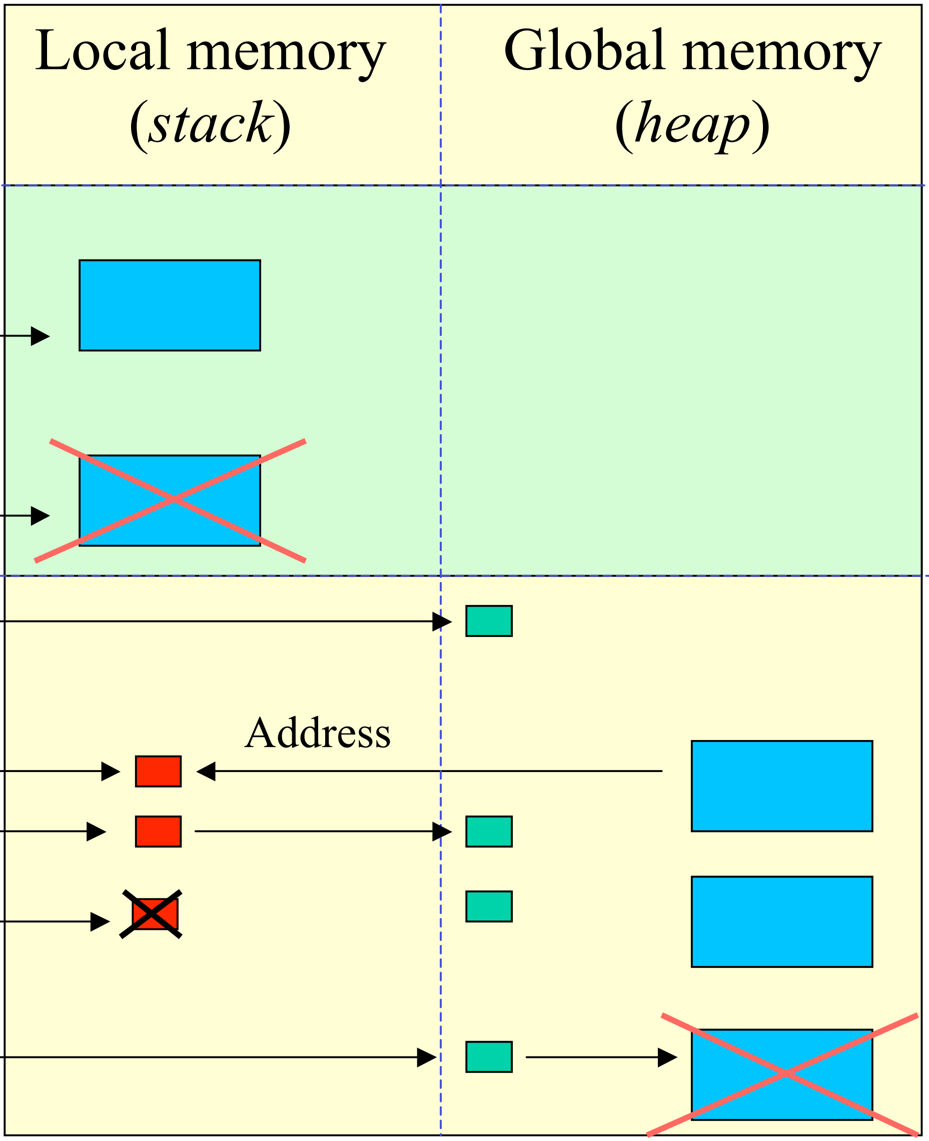
The C++ for ROOT cheat sheet

- Scope problems

```
void toto1(void)
    {
    house AHouse;
    AHouse.SetColour(red);
    …
    …
    }
```

```
house *MyHouse;
void toto2(void)
    {
    house* H_ptr=new house;
    MyHouse = H_ptr;
    H_ptr->SetColour(red);
    }
…

delete MyHouse;
```



The C++ for ROOT cheat sheet

- ROOT-specific details
  - all ROOT classes start with 'T' : `TVector,TH1F,TLine`
  - all ROOT constants start with 'k' : `kRed,kTRUE`
  - basic variable types are redefined (platform-independent), start with upper case, end in "`_t`" : `Double_t, Int_t`
  - informations about class members/methods :
    - interpreter command ".class" : `.class TLine`
    - using <TAB> on the command line:

      ```
      TLine l(0,0,1,1)
      l.Set<TAB>
      ```

    - using the method **`DrawClass()`** :

      ```
      l.DrawClass()
      ```

    - by internet : `http://root.cern.ch`