Samenvatting PVM

# Declarations And Definitions

! C++ uses single pass compilation

* Single Pass Compilation
  + Read in a bit of code
  + Check the code for correctness (syntax, types…)
  + Produce assembly code
  + Forget as much as possible (Free up RAM)
  + Proceed to the next bit of code

Problem :

double foo(){

return bar;

}

double bar(){

return 5;

}

* Does not compile : function foo does not yet know function bar

Solutions :

* Put the bar function before the foo function
* Forward Declaration : Provide the compiler with minimal though sufficient information

Bv :

int foo(double x);

int bar(bool x);

int foo(double x) {

return bar(false);

}

Int bar(bool x){

return foo(0);

}

! A declaration provides partial information and is a promise , A definition provides all information.

Example Class Declaration/Definitions:

// Declaration

Class Counter {

private :

int x;

public :

Counter();

void increment();

int read();

};

// Member function definitions

Counter::Counter() : x(0) { }

void Counter::increment() {x++;}

int Counter::read() { return x; }