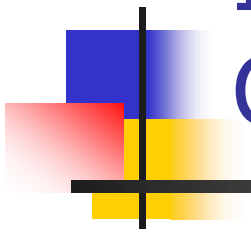


Introduction to Computer Programing
C++ Ch4 Important Notes

National Taiwan University
Dept. of Chemical Engineering
Prof. Chengche Hsu

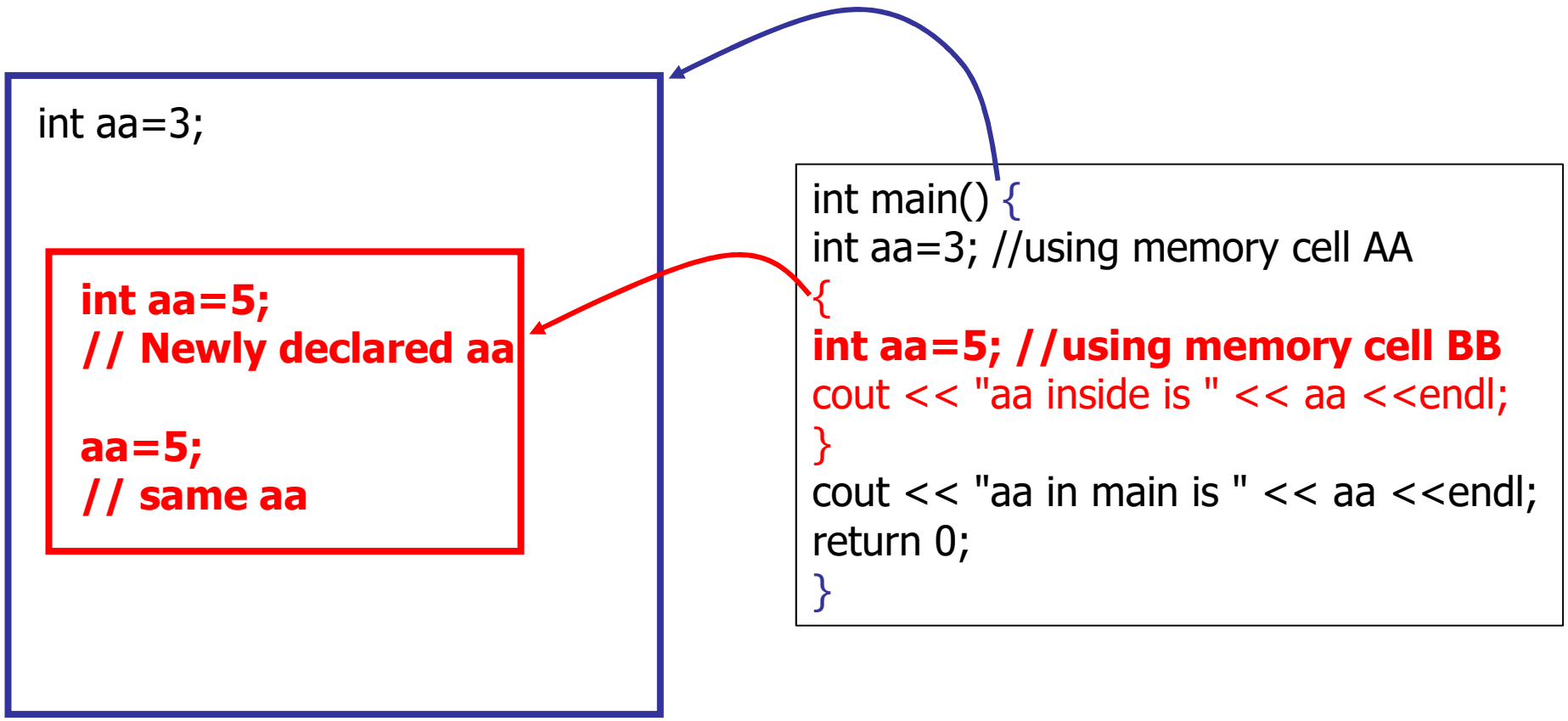
Scope and Lifetime of Variables

- The scope of a variable is the portion of the program in which statements can use that variable to refer to that particular memory cell.
- The scope of a function parameter and local variables is limited to the body of that function.
- The storage of local variables is created in at the entry of {} and destroyed at the exit of {}.

```
int main(){
int a=3; //using memory cell AA
{
int a=5; //using memory cell BB
cout << "a inside is " << a <<endl;
}
cout << "a in main is " << a <<endl;
return 0;
}
```

```
int main(){
int a=3; // using memory cell AA
{
a=5; // using memory cell AA
//a =3 is replaced by 5.
cout << "a inside is " << a <<endl;
}
cout << "a in main is " << a <<endl;
return 0;
}
```

Scope of Variables



Variable Scopes – More Examples

```
int main(){
    int a=2;
    cout << “initial a is " <<a<<endl;
    if (a!=1) {
    int a=4;
    cout<< "inside, a is "<<a<<endl;
    }
    cout<< "after, a is "<<a<<endl;
return 0;
}
```

```
int funtest(int);

int main(){
            int b, a=2;
    cout << "a before funtest: "<<a<<endl;
    b=funtest(2);
    cout << "a after funtest: "<<a<<endl;
return 0;
}

int funtest(int a){
            a=a*4;
            return a;
}
```

Global Variable

Declaration of a global variable

A global variable can be accessed even if a local variable of the same name is declared.

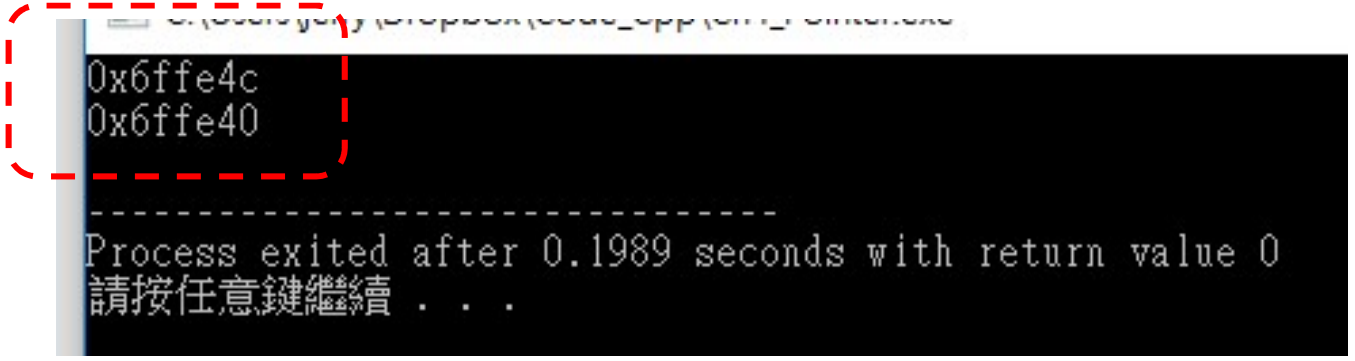
```
#include <iostream>
#include <cmath>
using namespace std;

double b=2;

int main(){
        double b=100;
    cout << "Local b: " << b << endl;
    cout << "Global b: " << ::b << endl;
return 0;
}
```

Pointer and Address

A pointer is a special type of data that stores the (memory) "address" of a variable.

Getting address in which aa or bb is stored.

```
int main(){
        int aa=2;
        double bb=4;
    cout << &aa << endl;
    cout << &bb << endl;
return 0;
}
```



Declaration of a Pointer

To declare a pointer:
Syntax:

```
int *aa_p, *bb_p;
```

To store the address to a pointer:
Syntax:

```
aa_p=&aa;
bb_p=&bb;
```

```
*aa_p=3;
*bb_p=3;
```

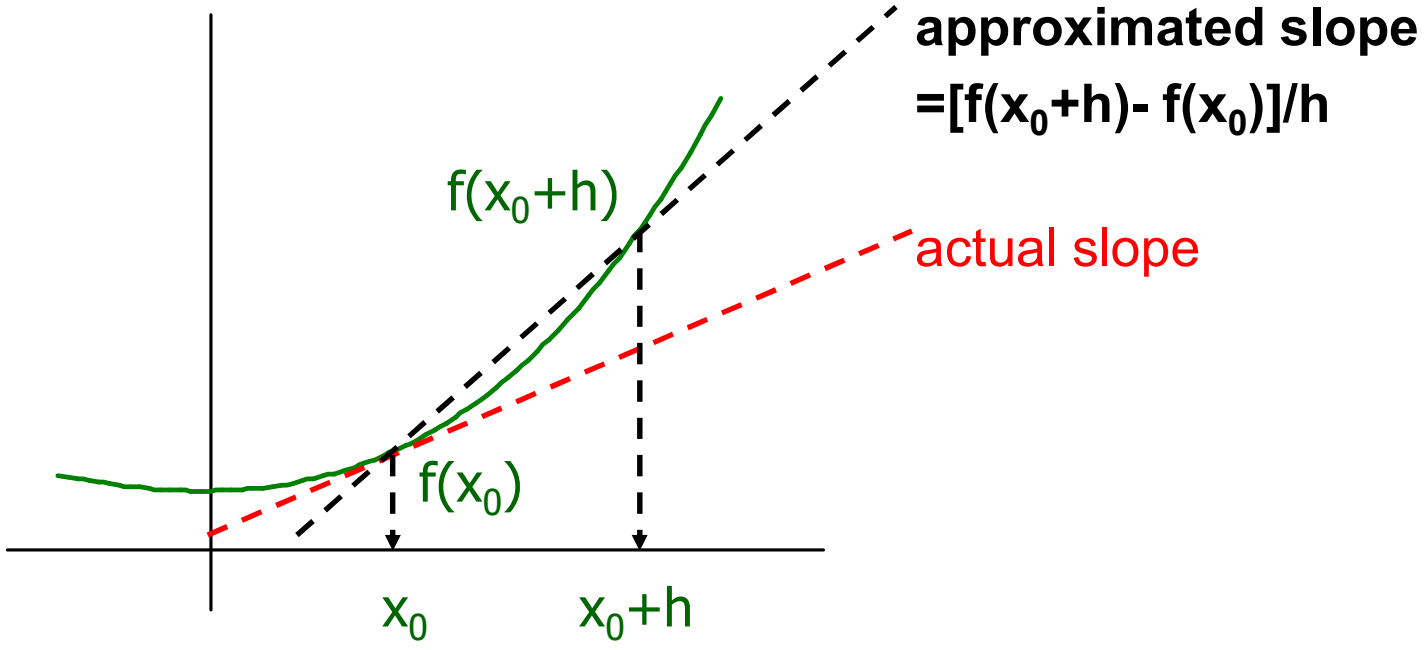
is equivalent to

```
aa=3;
bb=3;
```

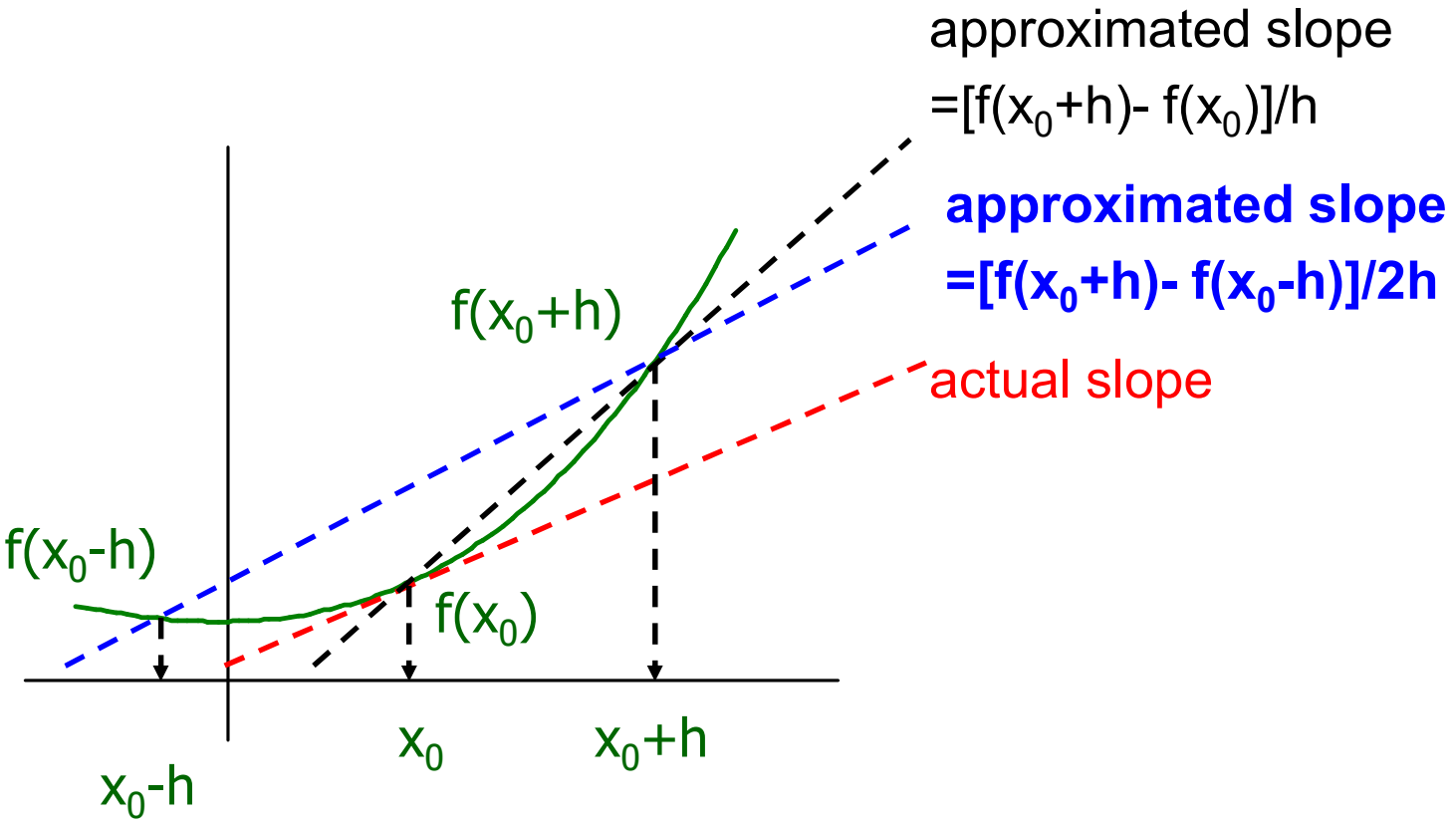
```
int main(){
    int aa;
    int *aa_p; // declare a pointer
    aa_p=&aa; // memory of aa assignment
    *aa_p=3; // store 3 into memory space
    cout << "aa is " << aa<< endl;
    cout << "*aa_p is "<< *aa_p << endl;
    cout << "aa_p is"<< aa_p << endl;
return 0;
}
```

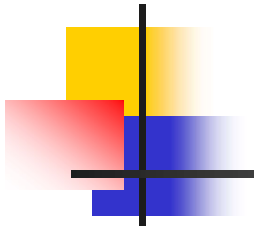
Note:
aa_p: the address
*aa_p: the value stored in the address (aa)

Finding Derivatives: Method 1



Finding Derivatives: Method 2





Finding Derivatives: More about Functions

1. Create a function that does f(x)=…. (defined inside the function, for example: $y=x^3-5x^2+3x+5-\exp(x)/3.$)
2. Create another function that inputs x0 and h, then calculates (numerically) f’(x) using method 1
3. Do the same thing using method 2
4. Using these two function to find the real root(s)