

Lab #07

Date: 2014/04/16

Agenda

- Recall class concept
- static
- class Pointer
- *this* pointer in class
- copy constructor

Class

- Recall that class has two type of members:
 - Data members - where you store the data
 - Member functions - define what the class can do

Now we illustrate the class concept by...



We suppose ...

- ATM only has three function:
 - Deposit: put the money in a bank
 - Withdraw: take the money from the bank
 - ChangePWD: change the password of user
- Each function would check the account and balance
- And the constructor and destructor can take the user balance out and put back after doing certain function (強強強constructor跟強強強destructor)

Pseudo code

- ```
void deposit(int amount) {
 if you deposit the correct amount:
 balance += amount;
 else:
 print "illegal operation!";
}
```
- ```
void withdraw(int amount) {  
    if balance >= amount:  
        give money equals "amount";  
        balance -= amount;  
    else:  
        print "not enough balance";  
}
```

Pseudo code

- ```
bool changePWD(int oldPWD[], int newPWD[]) {
 check the identity;
 if oldPWD == PWD:
 PWD = newPWD;
 else:
 print "illegal operation";
}
```

# Overview

- A user is in ...

- Main{

```
User Pusheen = "meow";
```

```
ATM citi(Pusheen);
```



## what will happen next?

- And ATM is a class defined here

- class ATM{

```
public:
```

```
int balance;
```

```
int PWD[10];
```

```
void deposit(int amount) ;
```

```
void withdraw(int amount) ;
```

```
bool changePWD(int oldPWD[],
int newPWD[]);
```

```
};
```

```
}
```



# what will happen next?

- The richest cat in the world is here
- Main{  
    User Pusheen = "meow";  
    ATM citi(Pusheen);  
  
    citi.balance = 999999999999;  
  
}



# Then...

```
• Main{
 User Pusheen = "meow";
 ATM citi(Pusheen);

 citi.balance = 999999999999

 User KoKo= "bark";
 srand(time(NULL));
 for (int i; i<10; i++) citi.PWD[i] = rand();
}
```



# That why we need private member in class

```
class ATM
{
 private:
 int balance;
 int PWD[10];
 public:
 void deposit(int amount) ;
 void withdraw(int amount) ;
 bool changePWD(int oldPWD[], int newPWD[]);
};
```

# Pointer to class

```
int main()
{
 ATM a(Pusheen);
 ATM* ptr = &a;
 a.deposit(100);
 ptr->deposit(100);
 (*ptr).deposit(100);

 return 0;
}
```

What is the executing result?

**A: 300**

# this pointer

- *this* pointer is an implicit pointer in Class
- It is the pointer point to the object itself
- In most cases, you don't need to write *this* pointer

Ex.

```
void deposit(int amount) {
 balance += amount;
}
```

is as same as

```
void deposit(int amount) {
 this->balance += amount;
}
```

# this pointer

- You need to have this pointer in two cases:
  - The parameter name is same as the data member name

```
void deposit(int balance) {
 this->balance += balance;
}
```

# this pointer

- You want to return object

```
ATM copy() {
 return *this;
}
```

```
int main() {
 ATM a, b;
 a.balance = 100;
 b = a.copy();
 cout << b.balance; //100
 return 0;
}
```

//this example will have some problems

//will be discussed later in ~~this semester~~ few minutes

# Also recall that...

- There are three necessary components in a class:
  - Constructor (建構子)
  - Destructor (解構子)
  - Copy constructor (複製建構子)



# If Pusheen wants to create account with the same balance in another bank...

- ```
Main{  
    User Pusheen = "meow";  
    ATM citi(Pusheen);  
    ATM hsbc = citi;
```

What wrong will happen?

```
}
```

When Pusheen change its password in citibank...

```
• Main{  
    User Pusheen = "meow";  
    ATM citi(Pusheen);  
    ATM hsbc = citi;  
  
    citi.changePWD(oldPWD, newPWD);  
    //the password of citibank and hsbc are the newPWD  
    //now!!  
}
```

Why?

- Because the default copy constructor do this:

```
ATM::ATM(const ATM& b)
{
    balance = b.balance;
    PWD = b.PWD ; // PWD[0]'s address
}
```

Copy constructor

- When do we copy an object?
 - When pass an object into a function using the call-by-value mechanism.
 - When we assign an object to another object.
 - When we create an object with another object as the argument of the constructor.

So you need to implement a deep copy version of copy constructor

```
ATM::ATM(const ATM& b)
{
    balance = b.balance;
    for (int i; i < 10; i++)
        PWD[i] = b.PWD[i] ;//PWD[0]'s address
}
```

friend

- One class can allow its “friends” to access its private members
- If there is a internet banking system to access the ATM class data member

- ```
class ATM{
 //.....
 friend class WebBank;
}; //Correct
```

- ```
class WebBank{  
    //.....  
    friend class ATM;  
}; //Wrong
```

Fun time!

- Have you heard about Shakespeare?
- Yes, you're right! It is of course a programming language!
- Is it special?

A character

Act I: Hamlet's insults and flattery.

Scene I: The insulting of Romeo.

[Enter Hamlet and Romeo]

Hamlet:

You lying stupid fatherless big smelly half-witted coward! $(x = -64)$

You are as stupid as the difference between a handsome rich brave
hero and thyself! Speak your mind! $(x = 8 - (-64); \text{print } x)$

Writing code: Hello world!

The Infamous Hello World Program.

Romeo, a young man with a remarkable patience.
Juliet, a likewise young woman of remarkable grace.
Ophelia, a remarkable woman much in dispute with Hamlet.
Hamlet, the flatterer of Andersen Insulting A/S.

Act I: Hamlet's insults and flattery.

Scene I: The insulting of Romeo.

[Enter Hamlet and Romeo]

Hamlet:
You lying stupid fatherless big smelly half-witted coward!
You are as stupid as the difference between a handsome rich brave
hero and thyself! Speak your mind!

You are as brave as the sum of your fat little stuffed misused dusty
old rotten codpiece and a beautiful fair warm peaceful sunny summer's
day. You are as healthy as the difference between the sum of the
sweetest reddest rose and my father and yourself! Speak your mind!

You are as cowardly as the sum of yourself and the difference
between a big mighty proud kingdom and a horse. Speak your mind.

Speak your mind!

[Exit Romeo]

Scene II: The praising of Juliet.

[Enter Juliet]

Hamlet:
Thou art as sweet as the sum of the sum of Romeo and his horse and his
black cat! Speak thy mind!

[Exit Juliet]

Scene III: The praising of Ophelia.

[Enter Ophelia]

Hamlet:

Thou art as lovely as the product of a large rural town and my amazing
bottomless embroidered purse. Speak thy mind!

Thou art as loving as the product of the bluest clearest sweetest sky
and the sum of a squirrel and a white horse. Thou art as beautiful as
the difference between Juliet and thyself. Speak thy mind!

[Exeunt Ophelia and Hamlet]

Act II: Behind Hamlet's back.

Scene I: Romeo and Juliet's conversation.

[Enter Romeo and Juliet]

Romeo:
Speak your mind. You are as worried as the sum of yourself and the
difference between my small smooth hamster and my nose. Speak your
mind!

Juliet:
Speak YOUR mind! You are as bad as Hamlet! You are as small as the
difference between the square of the difference between my little pony
and your big hairy hound and the cube of your sorry little
codpiece. Speak your mind!

[Exit Romeo]

Scene II: Juliet and Ophelia's conversation.

[Enter Ophelia]

Juliet:
Thou art as good as the quotient between Romeo and the sum of a small
furry animal and a leech. Speak your mind!

Ophelia:
Thou art as disgusting as the quotient between Romeo and twice the
difference between a mistletoe and an oozing infected blister! Speak
your mind!

[Exeunt]



Any Question?