

Week 6-2 : Constructor, Destructor

Copy Constructor

Part3. Copy Constructor

- Copy Constructor

When you recall name which is generated in parameter, copy the object.

If you are not definite copy constructor, default copy constructor insert automatically.

- Kinds of copy constructor through conversions

implicit conversion : = , explicit conversion : (object)

you have to use explicit to prevent implicit conversion

- Time of calling copy constructor

1. Initialize a new object using a already generated object.

```
Point x2(x1);
```

2. Call-by-value : pass the object as a parameter during the function calling

```
Point copyFunc(Point obj)
```

```
{  
    return obj;  
}
```

3. return the object which is not returned by the references.

```
Point copyFunc(Point obj)
```

```
{  
    return obj;  
}
```

**M1522.000600 Computer Programming
(2015 Spring)**

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

class Book
{
private:
char * bookName;
int bookNum;
public:
Book(char * tempName, int tempNum)
{
int len=strlen(tempName)+1;
bookName=new char[len];
strcpy(bookName, tempName);
bookNum=tempNum;
}

void ShowBookInfo() const
{
cout<<"Book Name : "<<bookName<<endl;
cout<<"Book Number : "<<bookNum<<endl;
}

~Book()
{
delete []bookName;
cout<<"destructor"<<endl;
}
};

int main(void)
{
Book book1("Computer Programming", 2001001);
Book book2("This is C++", 400010);
Book book3(book2);
book1.ShowBookInfo();
book2.ShowBookInfo();
book3.ShowBookInfo();
return 0;
}
```

```
}  
}
```

if you not define any copy constructor, a default copy constructor copies member to member. Upper code has an error, the default copy constructor points same book name part, destructor destruct at book2, and destructor destruct at book3, too. But there is nothing to destruct. because string already destructed. so, error appears.

To solve this problem, it needs to copy this book name part into another memory. this is called "deep copy".

```
Book(const Book& copy) : bookNum(copy.bookNum)  
{  
    bookName = new char[strlen(copy.bookName)+1];  
    strcpy(bookName, copy.bookName);  
}
```

[Exercise]

1. the main function below is a code that check the copy constructor. Make a Point Class and function. Copy constructor have to made by getX, getY.

```
class Point {
private:
int x;
int y;
public:
// Constructor

// Destructor

// Copy Constructor

void showXY();
void move(int _x, int _y);
int getX() const {return x;}
int getY() const {return y;}
};
```

```
int main() {
    Point p(1,2);
    Point cp(p);
    Point cp1=p;

    Point *p1=new Point(2,3);
    Point *cp2=new Point(p1);

    cout<<"p.showXY";
    p.showXY();
    p.move(3,4);
    cout<<"p.showXY";
    p.showXY();

    cp.move(4,5);
    cout<<"p.showXY";
    p.showXY();
    cout<<"cp.showXY";
    cp.showXY();
    cout<<"cp1.showXY";
    cp1.showXY();

    p1->move(2,3);
    cp2->move(-2,-3);
    cout<<"p1.showXY";
    p1->showXY();
    cout<<"cp2.showXY";
    cp2->showXY();
    delete p1,cp2;
    system("PAUSE");
    return EXIT_SUCCESS;
}
```