

Week 6-1 : Constructor, Destructor

Copy Constructor

Part1. Constructor

- Form of the Constructor

Get a same name with class name

there is no return type, truly return nothing.

One of the function, possibly set the overloading and default value.

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

class Constructor
{
int num1;
int num2;

public:
Constructor()
{
num1=0;
num2=0;
}
Constructor(int n)
{
num1=n;
num2=0;
}
Constructor(int n1, int n2)
{
num1=n1;
num2=n2;
}
```

```
}

/* default parameter constructor
Constructor(int n1=0, int n2=0)
{
num1=n1;
num2=n2;
}
*/

void ShowData() const
{
cout<<num1<<' '<<num2<<endl;
}
};

int main(void)
{
Constructor sc1;
sc1.ShowData();

Constructor sc2(100);
sc2.ShowData();

Constructor sc3(100, 200);
sc3.ShowData();
return 0;
}
```

When sc1, sc2, sc3 objects are being made, they pass overloaded constructor. If you use the default parameter constructor, then you erase other constructors. the result is same.

- The initialization using member initializer

Use member initializer when you call constructors of the objects which is declared as member variable.

Not initialize at the body, initialize at the next of the parameters.

```
#include <iostream>
```

```
using namespace std;

class Constructor
{
int num1;
int num2;

public:

Constructor(int n1, int n2) : num1(n1), num2(n2)
{
}

void ShowData() const
{
cout<<num1<<' '<<num2<<endl;
}
};

int main(void)
{
Constructor sc(100,200);
sc.ShowData();

return 0;
}
```

Part2. Destructor

Destruct the resources which is allocated by constructor.

If there is memory space allocated by new operator, then destructor destruct this memory space.

reference>> new and delete

They are compared to malloc and free respectively.

When you generate objects, you have to use "new".

**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);
book1.ShowBookInfo();
book2.ShowBookInfo();
return 0;
}
```

[Exercise]

1. In the first example, Think about generating sc1 object using 'Simpleclass sc1();' instead of 'Simpleclass sc1;'.
2. Make a member manager program.
 - Member class includes private name, age, department variables.
 - Initialize objects using constructor, also make a destructor.
 - Make a public set() method ; use this method, when you input member's information.
 - In the main, make a person type array object which has 3 indexes.
 - Input the member's information using for loop.