13 Dynamic Object Creation

Hu Sikang

skhu@163.com

13.1 Object creation

- When a C++ object is created, two events occur:
  - Storage is allocated for the object.
    - static storage area
    - stack (Local variable memory allocation)
    - heap (dynamic memory allocation)
  - The constructor is called to initialize that storage.

operator new/delete

```
new type(initializer);  // create a dynamic object
delete pointername;   // destroy a dynamic object
```

- MyClass *fp = new MyClass(1,2);
- MyClass *fp = new MyClass;
- delete fp;

```
new type[size];   // create an array of objects
delete[] pointer; // destroy an array of objects
```

- MyClass *fp = new MyClass[100];
- delete [] fp;

13.2 Overloading new & delete

```
void* operator new(size_t size)  {
    void *t = malloc(size);
    NUM++;
    // Here constructor is called if type is CLASS.
    return t;
}
void operator delete(void* p) {
    // Firstly destrctor is called if type is CLASS
    NUM--;
    if (p != nullptr)    {
    // Avoid using cout because global operator new is called when creating cout.
    printf("%d, %d\n", *(int*)p, NUM);
    free(p);
    }
}
```

```
#include <stdio.h>
#include <malloc.h>
int NUM = 0;
void main()
{
    int *a = new int(10);
    int *b = new int(20);
    int *c = new int(30);

    delete a;
    delete b;
    delete c;
}
```

13.2 Overloading new & delete

When does it need to be overloaded operator new & delete?

- **When special codes need to be run in operator new & delete.**
- **When huge storage space should be allocated in the disk as virtual memory space.**