1

```
#include <iostream>
using namespace std;
class A
{
public:
        A(int n)
        {
                if (n <= 0) {
                        throw "caught in the main";
                }
                cout << "ID=" << n << ": constructed₩n";
                n_ID = n;
                data = new int[n];
        }
        ~A()
        {
                cout << "ID=" << n_ID << ": destroyed₩n";
        }
private:
        int* data = NULL;
        int n_ID;
};
int main() {
        try {
                A a(3);
                A b(2);
                {
                        A c(1);
                        A d(0);
                        A e(-1);
                }
        }
        catch (const char* msg) {
                cerr << msg << endl;
        }
        return 0;
}
```

2

```
#include <iostream>
using namespace std;
int data_size = 5;
class A {
public:
        A() {
                data = new int[data_size];
                for (int i = 0; i < data_size; i++)
                        data[i] = i;
                cout << "constructed₩n";
        }
        A(const A& a) {
                data = new int[data_size];
                for (int i = 0; i < data_size; i++) {
                        data[i] = i;
                }
        }
        ~A()
        {
                for (int i = 0; i < data_size; i++)
                        data[i] = 0;
                delete[] data;
                data = NULL;
                cout << "destroyed₩n";
        }
private:
        int* data = NULL;
        friend ostream& operator <<(std::ostream& os, const A& a);
};
ostream& operator <<(std::ostream& os, const A& a)
{
        for (int i = 0; i < data_size; i++)
                os << a.data[i] << " ";
        return os;
}
int main()
{
        try
        {
                A a;
                cout << a << endl;
                throw a;
        }
        catch (A& a)
        {
                cout << "err. handled₩n";
                cout << a << endl;

        }
        return 0;
}
```