

C++与C的主要差异

– 动态内存分配和释放程序实例

【例1-9】使用堆内存分配方式实现学生成绩录入功能，要求程序运行时由用户输入学生人数。

```
#include <iostream>
using namespace std;
int main()
{
    int *pScore;
    int n, i;
    cout<<"请输入学生人数:";
    cin>>n;
    pScore=new int[n];//按学生人数动态分配内存
    if (pScore==NULL)//判断堆内存分配是否成功
    {
        cout<<"堆内存分配失败!"<<endl;
        return 0;
    }
```

```
//通过for循环输入学生成绩
for (i=0; i < n; i++)
{
    cout<<"请输入第"<<(i+1)
        <<"名学生的成绩:";
    cin>>pScore[i];
}
//通过for循环输出学生成绩
for (i=0; i < n; i++)
    cout<<"第"<<(i+1)
        <<"名学生的成绩为: "
        <<*(pScore+i)<<endl;
//不再使用时将堆内存及时释放
delete []pScore;
return 0;
}
```

【例1-10】使用堆内存分配方式实现学生信息录入功能，要求程序运行时由用户输入学生人数。

```
#include <iostream>
using namespace std;
struct Student
{
    char num[8];
    char name[10];
    int score;
};
int main()
{
    Student *pStu;
    int n, i;
    cout<<"请输入学生人数:";
    cin>>n;
    //根据学生人数动态分配内存
    pStu=new Student[n];
```

```
    if (pStu==NULL)//判断堆内存分配是否成功
    {
        cout<<"堆内存分配失败!"<<endl;
        return 0;
    }
    for (i=0; i < n; i++)//通过for循环输入学生信息
    {
        cout<<"请输入第"<<(i+1)
            <<"名学生的学号、姓名和入学成绩:";
        cin>>pStu[i].num>>pStu[i].name
            >>pStu[i].score;
    }
    for (i=0; i < n; i++)//通过for循环输出学生信息
        cout<<"第"<<(i+1)
            <<"名学生的学号、姓名和入学成绩为:"
            <<(pStu+i)->num<<"'<<(pStu+i)->name
            <<"'<<(pStu+i)->score<<endl;
    delete []pStu;//不再使用时将堆内存及时释放
    return 0;
}
```