C++ Memory Management: new and delete

Dynamic Memory Allocation and Deallocation

Program 1

```
#include<iostream.h>
using namespace std;
int main()
{
        int    *ptr_i =new int(25);
        float  *ptr_f =new float(10.1234);
        char   *ptr_c =new char('a');
        double *ptr_d =new double(1234.566);
        cout<<"\n\n\n\nContents of the pointers "<<endl;
        cout<<"Integer ="<<*ptr_i<<endl;
        cout<<"Float ="<<*ptr_f<<endl;
        cout<<"Char ="<<*ptr_c<<endl;
        cout<<"Double ="<<*ptr_d<<endl;

        delete ptr_i;
        delete ptr_f;
        delete ptr_d;
        delete ptr_c;

}
```

Program 2:

```
#include<iostream>
using namespace std;
int main()
{
        int *ptr_a=new int;
        int *ptr_b=new int;
        int *pt_sum=new int;
        int *pt_sub=new int;
        int *pt_mult=new int;
        int *pt_div=new int;
        cout<<"Enter any two integers"<<endl;
        cin>>*ptr_a>>*ptr_b;

        *pt_sum = *ptr_a + *ptr_b;
        *pt_sub = *ptr_a - *ptr_b;
        *pt_mult = *ptr_a * *ptr_b;
        *pt_div = *ptr_a / *ptr_b;

        cout<<"Addition ="<<*pt_sum<<endl;
        cout<<"Substraction ="<<*pt_sub<<endl;
        cout<<"Multiplication ="<<*pt_mult<<endl;
        cout<<"Division ="<<*pt_div<<endl;
        delete ptr_a;
        delete ptr_b;
        delete pt_sum;
        delete pt_sub;
        delete pt_mult;
        delete pt_div;
        }
```

Program 3:

```
#include<iostream >
using namespace std;
int main()
{
	clrscr();
	int *ptr_a=new int[20];   // int num[20];
	int *ptr_n=new int;

	cout<<"\n\n\n\nHow many numbers are there"<<endl;
	cin>>*ptr_n;
	for(int i=0 ; i < *ptr_n ; ++i)
	{
		cout<<"Element :";
		cin>>ptr_a[i];			// num[i]
	}
	cout<<"Contents of the array\n";
	for(i=0;i<*ptr_n;i++)
	{
		cout<<ptr_a[i];
		cout<<"\t";
	}
	delete ptr_n;
	delete [ ] ptr_a;

}
```

Program 4:

```
#include<iostream>
using namespace std;
class Sample
{
	private:
		int x; float y;
	public:
```

```
                void getdata();
                void display();
};
inline void Sample::getdata()
{
        cout<<"\n\n\n\nEnter an integer value\n";
        cin>>x;
        cout<<"\n\nEnter a floating point value\n";
        cin>>y;
}
inline void Sample::display()
{
        cout<<"\n\n\n\nEntered numbers are\n";
        cout<<"X = "<<x<<"\t"<<"Y= "<<y<<endl;
}
int main()
{

        Sample *ptr;                    // Sample *ptr = new Sample();
        ptr=new Sample;
        ptr->getdata();
        ptr->display();
        delete ptr;

}
```