new/delete in C++

Example1:

```
#include<iostream>
using namespace std;
int main()
{
	//int *p1 = new int;
	int *p1;
	p1 = new int;
	*p1 = 1000;
	int *p2 = new int(2000); // Another way of allocating using new.
	cout<<"The first value is :"<<*p1<<endl;
	cout<<"The second value is :"<<*p2<<endl;
	delete p1;
	delete p2;
	return 0;
}
```

Example2:

```
// Initializing array dynamically
#include<iostream>
using namespace std;
int main()
{
	int *p = new int[5]; // allocate an array of size 5 (integer type)
	cout<<"Enter the values of the array:"<<endl;
	for(int i =0;i<5;i++)
	{
		cin>>p[i];
	}
	cout<<"The elements of the array are:"<<endl;
	for(int i =0;i<5;i++)
	{
		cout<<p[i]<<endl;
	}
	delete p[];
	return 0;
}
```

Example3:

```
#include<iostream>
using namespace std;
struct st
{
	int a, b;
	void display()
	{
		cout<<a<<","<<b<<endl;
	}
};
int main()
{
	st *s = new st;
	s->a = 10;
	s->b = 40;
	s->display();
	delete s;
}
```

Example 4:

```
#include<iostream>
#include<string.h>
using namespace std ;
class employee
{
	private:
		char name[20];
		int age;
		float sal;
	public:
		employee()
		{
			cout<<endl<<"Reached the zero-argument constructor";
			strcpy(name,"");
			age = 0.0;
			sal = 0.0;
		}

		employee(char *n, int a, float s)
		{
			cout<<endl<<"Reached the three argument constructor";
			strcpy(name,n);
			age = a;
			sal = s;
		}
		void setdata(char *n, int a, float s)
		{
			strcpy(name,n);
			age = a;
			sal = s;
		}
		void showdata()
		{
			cout<<endl<<name<<"\t"
					<<age<<"\t"
					<<sal;
		}

		~employee()
		{
			cout<<endl<<"Reached the destructor"<<endl;
		}
};
int main()
{
	employee *p;
	p = new employee;
	p->setdata("Ram",23,4500.50 );
	employee *q;
	q= new employee("Bharavt",24,3400.60);
	p->showdata();
	q->showdata();
	delete p;
	delete q;
	return 0;
}
```

Use of "static"

Example 1:

```
#include<iostream>
using namespace std;
struct st
{
	static int x, y;
	static void print()
	{
		cout<<x<<"," <<y<<endl;
	}

};

int st::x;
int st::y;

int main()
{
	st s;
	//s.x = 10;
	//s.y = 20;
	st::x = 10;
	st::y = 20;
	st s1;
	//s1.x=22;
	//s1.y=33;
	st::x = 22;
	st::y = 33;
	//s.print();
	//s1.print();
	st::print();
	st::print();
	return 0;
}
```