new and delete operators in C++ with printing values through constructor and destructor

In C++ programming language, there are two operators 1) **new** and 2) **delete**, which are used to manage the memory dynamically i.e. to create, delete the memory at run time (dynamically)

`new` is used to declare memory blocks at run time (dynamically). While, `delete` is used to delete/free the memory which has been declared dynamically.

Example of new and delete in C++

In the given program, we are using new to allocate memory to the class object and delete/free is using to delete the reference of the pointer that will force. There is a class named `sample` and it has a constructor `sample()` and a destructor `~sample()`.

Consider the program:

```
#include <iostream>
using namespace std;

class sample
{
    public:
      sample()
      {
            cout<<"Hi ";
      }
      ~sample()
      {
            cout<<"Bye ";
      }
};

int main()
{
      sample *obj = new sample();
      delete(obj);
      return 0;
}
```