```cpp
#include <iostream>
using std::cout;
using std::cin;
using std::endl;

class demo
{
    int data;
    static demo *p;

    demo()
    {
        data = 0;
    }

    ~demo()
    {
        data = 0;
    }

public:
    static demo* get_object()
    {
        if(NULL == p)
        {
            p = new demo;
            if(NULL == p)
            {
                cout << "Memory allocation FAILED\n";
                return NULL;
            }
        }
        else
        {
            bool ret;

            cout << "You can't create another object as this is singleton class\n";
            cout << "Are you want to use existing object? (1/0)\t";
            cin >> ret;

            if(ret == false)
                return NULL;
        }

        return p;
    }

    static void delete_object()         // this can be non-static
    {
        if(p != NULL)
        {
            delete p;
            p = NULL;
        }
    }

    void set_data(int param)
    {
        data = param;
    }

    void get_data()
    {
        cout << "Data is " << data << endl;
    }
};

```

```
68  demo* demo::p = NULL;
69
70  int main(void)
71  {
72  //  demo obj;                   // error
73
74      demo *p = NULL;
75
76      p = demo::get_object();
77      if(p != NULL)
78      {
79          p->get_data();
80          p->set_data(10);
81          p->get_data();
82      }
83
84  //  p->delete_object();         // or demo::delete_object();
85
86      p = demo::get_object();
87      if(p != NULL)
88      {
89          p->get_data();
90          p->set_data(20);
91          p->get_data();
92      }
93
94      return 0;
95  }
```