a) Write a class that has marks and grade as data members. A constructor with two parameters initializes data members with given values and member function show that displays the values of data members. Create two objects and display the values.

Student.cpp

```
class Student{
    private:
        char _grade;
        string _name;
        vector<ll int> _marks;
    public:
        // Constructor
        Student(string name, vector<ll int> marks, char grade) : _name(name), _marks(marks),
_grade(grade){
            printf("Object(name = %s) has been created.\n", _name.c_str());
        }

        // Setters
        void name(string name){
            _name = name;
        }
        void marks(vector<ll int> marks){
            _marks = marks;
        }
        void grade(char grade){
            _grade = grade;
        }

        // Getters
        string name(){
            return _name;
        }
        vector<ll int> marks(){
            return _marks;
        }
        char grade(){
            return _grade;
        }

        // Functionality
        void show(){
            printf("Student(name = %s, grade = %c)\n", _name.c_str(), _grade);
        }

        // Destructor
        ~Student(){
            printf("Object(name = %s) has been deallocated.\n", _name.c_str());
        }
};
```

b) Create a class TV that contains attributes of brand name, model, and retail price. Write a method to display all attributes and a method to change the attributes. Also write a method to initialize all the attributes.

TV.cpp

```
class TV{
    private:
        string _brand, _model;
        long double _price;
    public:
        // Constructor
        TV(string brand, string model, long double price) : _brand(brand), _model(model),
_price(price){
            printf("Object(brand = %s) has been allocated.\n", _brand.c_str());
        }

        // Setters
        void brand(string brand){
            _brand = brand;
        }
        void model(string model){
            _model = model;
        }
        void price(long double price){
            _price = price;
        }

        // Getters
        string model(){
            return _model;
        }
        string brand(){
            return _brand;
        }

        long double price(){
            return _price;
        }

        // Functionality
        void show(){
            printf("TV(brand = %s, model = %s, price = %.2lf)\n", _brand.c_str(),
_model.c_str(), _price);
        }

        // Destructor
        ~TV(){
            printf("Object(brand = %s) has been deallocated.\n", _brand.c_str());
        }
};
```

c) Write a class that has num and ch as data members. A constructor with no parameters initializes num to 0 and ch to ‘x’. A constructor with two parameters initializes data members with the given values and a member function show that displays the values of data members.

Student.exe

```
class Any{
    private:
        char _ch;
        ll int _num;
    public:
        // Constructor
        Any(ll int num = 0, char ch = 'x') : _num(num), _ch(ch){}

        // Setters
        void num(ll int num){
            _num = num;
        }
        void ch(char ch){
            _ch = ch;
        }

        // Getters
        ll int num(){
            return _num;
        }
        char ch(){
            return _ch;
        }

        // Functionality
        void show(){
            printf("Any(num = %ld, ch = %c)\n", _num, _ch);
        }

        // Destructor
        ~Any(){}
};
```

d) Write a class book that has attributes for pages, price, and title. It has two functions one to input the values and the other to display the values. Create three objects of the class with the same input values.

Book.cpp

```
#include <bits/stdc++.h>
using namespace std;

#define ll long long

class Book{
    private:
        string _title;
        ll int _pages, _price;
    public:
        // Constructor
        Book(string title, ll int pages, ll int price) : _title(title), _pages(pages),
_price(price){}

        // Setters
        void title(string title){
            _title = title;
        }
        void pages(ll int pages){
            _pages = pages;
        }
        void price(ll int price){
            _price = price;
        }

        // Getters
        string title(){
            return _title;
        }
        ll int pages(){
            return _pages;
        }
        ll int price(){
            return _price;
        }

        // Functionality
        void show(){
            printf("Book(title = %s, pages = %ld, price = %ld)\n", _title.c_str(), _pages,
_price);
        }

        // Destructor
        ~Book(){}
};
```