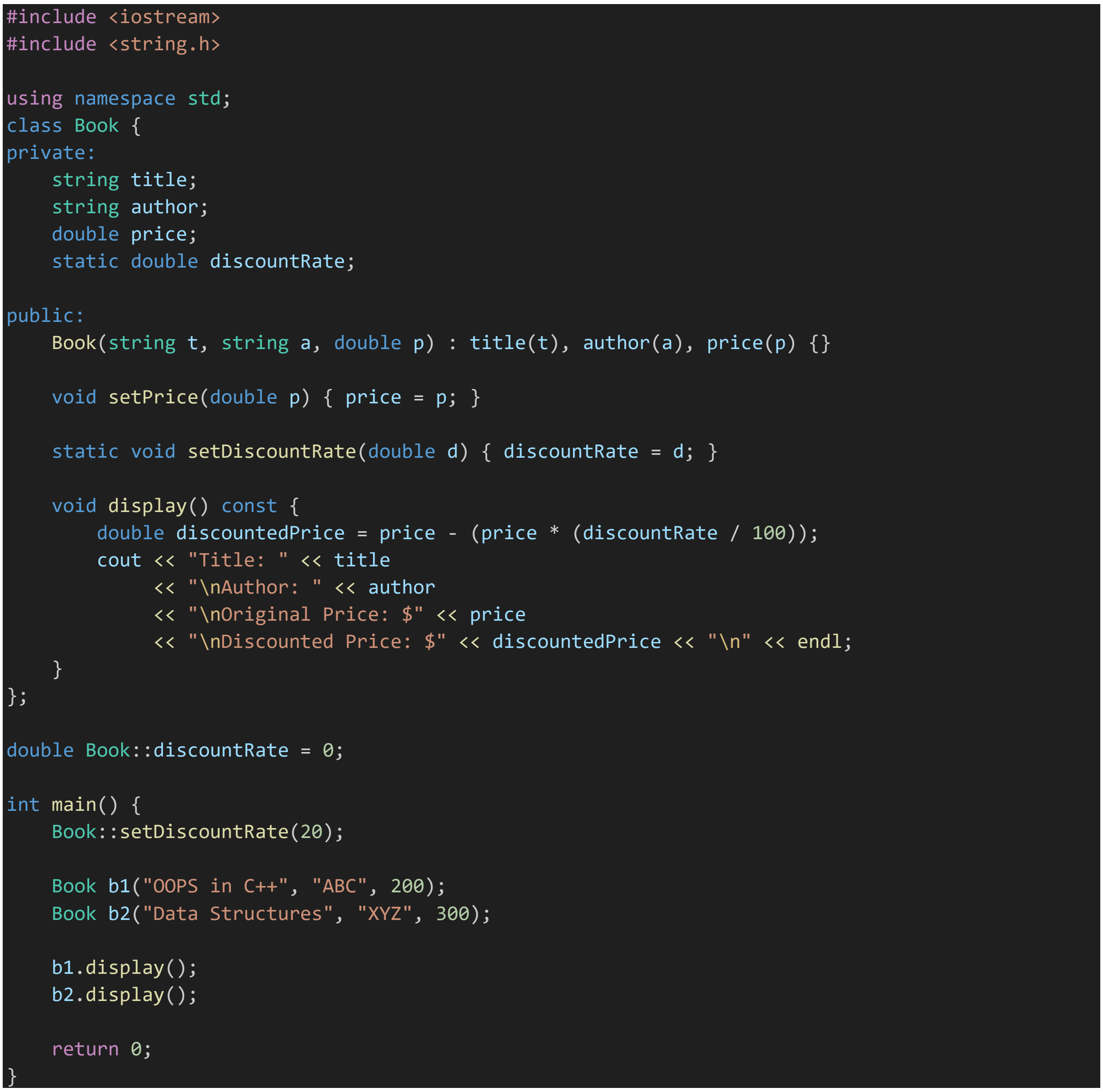
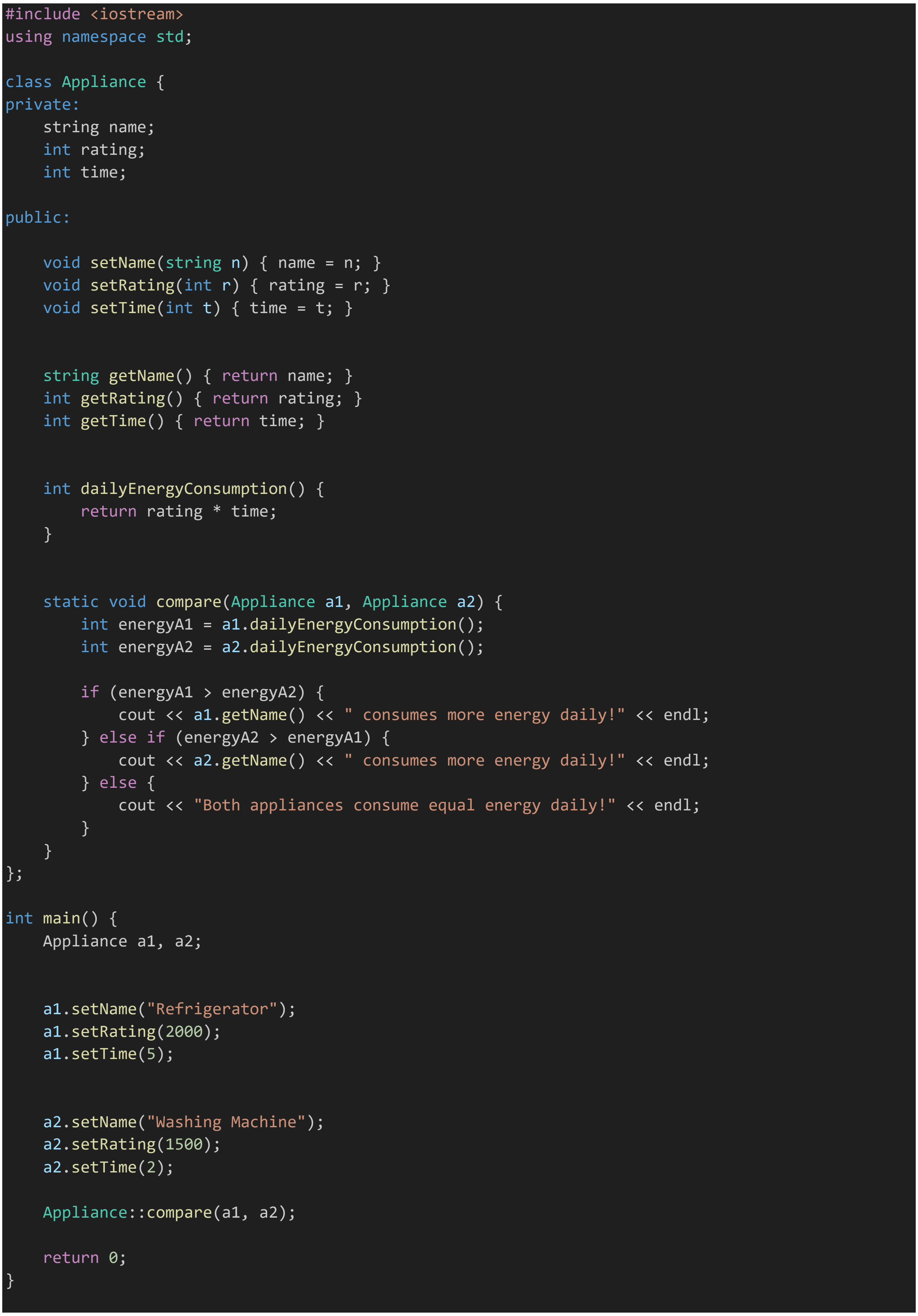
Lab 01:



```cpp
#include <iostream>
#include <string.h>

using namespace std;
class Book {
private:
    string title;
    string author;
    double price;
    static double discountRate;

public:
    Book(string t, string a, double p) : title(t), author(a), price(p) {}

    void setPrice(double p) { price = p; }

    static void setDiscountRate(double d) { discountRate = d; }

    void display() const {
        double discountedPrice = price - (price * (discountRate / 100));
        cout << "Title: " << title
             << "\nAuthor: " << author
             << "\nOriginal Price: $" << price
             << "\nDiscounted Price: $" << discountedPrice << "\n" << endl;
    }
};

double Book::discountRate = 0;

int main() {
    Book::setDiscountRate(20);

    Book b1("OOPS in C++", "ABC", 200);
    Book b2("Data Structures", "XYZ", 300);

    b1.display();
    b2.display();

    return 0;
}
```

Lab 02:



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

class Appliance {
private:
    string name;
    int rating;
    int time;

public:

    void setName(string n) { name = n; }
    void setRating(int r) { rating = r; }
    void setTime(int t) { time = t; }


    string getName() { return name; }
    int getRating() { return rating; }
    int getTime() { return time; }


    int dailyEnergyConsumption() {
        return rating * time;
    }


    static void compare(Appliance a1, Appliance a2) {
        int energyA1 = a1.dailyEnergyConsumption();
        int energyA2 = a2.dailyEnergyConsumption();

        if (energyA1 > energyA2) {
            cout << a1.getName() << " consumes more energy daily!" << endl;
        } else if (energyA2 > energyA1) {
            cout << a2.getName() << " consumes more energy daily!" << endl;
        } else {
            cout << "Both appliances consume equal energy daily!" << endl;
        }
    }
};

int main() {
    Appliance a1, a2;


    a1.setName("Refrigerator");
    a1.setRating(2000);
    a1.setTime(5);


    a2.setName("Washing Machine");
    a2.setRating(1500);
    a2.setTime(2);

    Appliance::compare(a1, a2);

    return 0;
}
```

Lab 03:

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

class Course {
private:
    string name;
    int course_id;
    float fee;
    static int totalCourses;

public:
    Course(string name, int id, float fee) { this->name = name; this->course_id = id; this->fee = fee; totalCourses++;}

    void setName(string n) {
        this->name = n;
    }

    string getName() {
        return this->name;
    }

    void displayCourse() {
        cout << "Course: " << this->name << endl;
    }

    static int getTotalCourses() {
        return totalCourses;
    }
};

int Course::totalCourses = 0;

int main() {
    Course c1("OOPS", 1, 200.0), c2("DSA", 2, 300), c3("Algorithms", 3, 400.0);
    cout << "Total No. of Courses: " << Course::getTotalCourses() << endl;
    return 0;
}
```

Lab 04:

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

class Stock
{
private:
    string companyName;
    float stockPrice;
    int availableShares;
    static float marketValue;

public:
    Stock(string name, float price, int shares)
    {
        this->companyName = name;
        this->stockPrice = price;
        this->availableShares = shares;
    }

    void market_Value()
    {
        marketValue = stockPrice * availableShares;
    }

    static float getMarketValue() {
        return marketValue;
    }

    void displayStock()
    {
        cout << "📈 " << companyName
             << " | Price: $" << stockPrice
             << " | Shares: " << availableShares << endl;
    }

    void setPrice(float p) { this->stockPrice = p; }

    void buyShares(int quantity)
    {
        if (quantity > availableShares)
        {
            cout << "Not enough shares available to buy." << endl;
            return; // Stop
        }

        availableShares -= quantity;

        if (quantity > 10)
            stockPrice *= 1.01; // Increment of 1%

        market_Value();
        cout << "✅ Bought: " << quantity << " shares of " << companyName << endl;
    }

    void sellShares(int quantity)
    {
        availableShares += quantity;

        if (quantity > 10)
```

```
            stockPrice *= 0.99; // Decrement of 1%

        market_Value();
        cout << "✅ Sold: " << quantity << " shares of " << companyName << endl;
    }
};

float Stock::marketValue = 0.0;

int main()
{
    Stock apple("Apple", 150.0, 100),
    tesla("Tesla", 200.0, 80);


    apple.displayStock();
    tesla.displayStock();
    cout << "🌍 Total Market Value: $" << Stock::getMarketValue() << "\n\n";

    // Buy Shares
    apple.buyShares(15);
    tesla.buyShares(10);

    cout << "\n📊 After transactions:\n";
    apple.displayStock();
    tesla.displayStock();
    cout << "🌍 Total Market Value: $" << Stock::getMarketValue() << "\n";

    return 0;
}
```

Lab 05:

```
#include <iostream>
#include <cstdlib> // rand()
using namespace std;

class Ticket
{
private:
    int ticketNumber;
    string passengerName;
    int seatNumber;
    static int soldTickets;

public:
    Ticket(string pN, int sN)
    {
        ticketNumber = rand() % 1000 + 1;
        passengerName = pN;
        seatNumber = sN;
        soldTickets++;
    }

    void display() {
        cout << "Ticket No: " << ticketNumber << ", Passenger: " << passengerName << ", Seat: "
<< seatNumber << endl;
    }

    static int getSoldTickets() {
        return soldTickets;
    }
};

int Ticket::soldTickets = 0;

int main() {

    Ticket t1("Alice", 12);
    Ticket t2("Bob", 15);

    t1.display();
    t2.display();

    cout << "Total Tickets Sold: " << Ticket::getSoldTickets() << endl;

    return 0;
}
```