8-2 小專案參考解答

```
#include <iostream>
#include <ctime>

using namespace std;

struct Item
{
    string name;
    double price;
    int quantity;
};

struct Date
{
    int year;
    int month;
    int day;
};

struct Time
{
    int hh;
    int mm;
    int ss;
};

struct Invoice
{
    Item itemList[100];
    int countOfItemList;
    Date date;
    Time time;

    double amount;

    double taxRate;
    double totalDue;
};

Invoice prepareInvoice(double taxRate)
{
    Invoice invoice{};

    // get & set current date/time
```

```
    time_t now = time(nullptr);
    tm localTime{};
    localtime_s(&localTime, &now);

    invoice.date.year = localTime.tm_year + 1900;
    invoice.date.month = localTime.tm_mon + 1;
    invoice.date.day = localTime.tm_mday;

    invoice.time.hh = localTime.tm_hour;
    invoice.time.mm = localTime.tm_min;
    invoice.time.ss = localTime.tm_sec;

    invoice.countOfItemList = 0;
    invoice.amount = 0.0;
    invoice.taxRate = taxRate;
    invoice.totalDue = invoice.amount * (1.0 + invoice.taxRate);

    return invoice;
}

void printInvoice(const Invoice& invoice)
{
    printf("Date: %d/%d/%d\n", invoice.date.year, invoice.date.month,
invoice.date.day);
    printf("Time: %d:%d:%d\n", invoice.time.hh, invoice.time.mm, invoice.time.ss);

    printf("------------------------------\n");
    printf("Items:\n");
    for (int i = 0; i < invoice.countOfItemList; ++i)
    {
        const Item& item = invoice.itemList[i];
        printf("- %s: $%.2f x %d\n", item.name.c_str(), item.price, item.quantity);
    }

    printf("------------------------------\n");
    printf("Amount: $%.2f\n", invoice.amount);
    printf("Tax Rate: %.2f%%\n", invoice.taxRate * 100);
    printf("Total Due: $%.2f\n", invoice.totalDue);
}

void scanItem(Invoice& invoice)
{
    string name;
    double price;

    cin >> name;
    cin >> price;

    for (int i = 0; i < invoice.countOfItemList; ++i)
    {
```

```
        if (invoice.itemList[i].name == name)
        {
            invoice.itemList[i].quantity += 1;
            invoice.amount += price;
            invoice.totalDue = invoice.amount * (1.0 + invoice.taxRate);
            return;
        }
    }

    Item newItem{name, price, 1};
    invoice.itemList[invoice.countOfItemList++] = newItem;
    invoice.amount += price;
    invoice.totalDue = invoice.amount * (1.0 + invoice.taxRate);
}

int main()
{
    double taxRate = 0.05;

    Invoice invoice = prepareInvoice(0.05);

    int n;
    cin >> n;
    for(int i=0; i<n; ++i)
    {
        scanItem(invoice);
    }
    printInvoice(invoice);

    return 0;
}
```

Revision #2
★Created 2026-04-20 02:10:29 UTC by huihui
Updated 2026-04-20 02:12:59 UTC by huihui