void other(void)
{
    using namespace debts;
    Person dg = {"Doodles", "Glister"};
    showPerson(dg);
    cout << endl;
    Debt zippy[3];
    int i;

    for (i = 0; i < 3; i++)
        getDebt(zippy[i]);

    for (i = 0; i < 3; i++)
    {
        showDebt(zippy[i]);
        cout << "Total debt: $" << sumDebts(zippy, 3) << endl;
        return;
    }
}

void another(void)
{
    using pers::Person;;

    Person collector = { "Milo", "Rightshift" };  
    pers::showPerson(collector);
    cout << endl;
}

First, main() using two using-declarations:

using debts::Debt;   // makes the Debt structure definition available
using debts::showDebt;  // makes the showDebt function available

Note that using-declarations just use the name; for example, the second one doesn't
describe the return type or function signature for showDebt; it just gives the name. (Thus, if a function were overloaded, a single using-declaration would import all the versions.)
Also, although both Debt and showDebt() use the Person type, it isn't necessary to