import any of the `Person` names because the `debt` namespace already has a using-directive including the `pers` namespace.

Next, the `other()` function takes the less desirable approach of importing the entire namespace with a using-directive:

```cpp
using namespace debts; // make all debts and pers names available to other()
```

Because the using-directive in `debts` imports the `pers` namespace, the `other()` function can use the `Person` type and the `showPerson()` function.

Finally, the `another()` function uses a using-declaration and the scope resolution operator to access specific names:

```cpp
using pers::Person;;
pers::showPerson(collector);
```

Here is a sample run of the program:

```
Goatsniff, Benny: $120
Glister, Doodles
Enter first name: Arabella  
Enter last name: Binx
Enter debt: 100
Enter first name: Cleve
Enter last name: Delaproux
Enter debt: 120
Enter first name: Eddie
Enter last name: Fiotox
Enter debt: 200
Binx, Arabella: $100
Delaproux, Cleve: $120
Fiotox, Eddie: $200
Total debt: $420
Rightshift, Milo
```

Namespaces and the Future