

```

package Ledger::File;
use strict;
use warnings;

sub new {
    my $class = shift;
    my $self = {};
    my %args = @_;

    bless $self => $class;

    die "Required attribute 'ledger_file' missing\n"
        unless $args{ledger_file};

    $self->ledger_file($args{ledger_file});

    $self;
}

sub ledger_file {
    my ($self, $ledger_file) = @_;

    if (defined $ledger_file) {
        $self->{ledger_file} = $ledger_file;
    }

    $self->{ledger_file};
}

```

```

package Ledger::File;
use strictures 2;
use Types::Standard qw(Str);

use Moo;
use namespace::clean;

has ledger_file => (is => 'ro', isa => Str, required => 1);

```

“Favor object composition over class inheritance.”

Design Patterns: Elements of Reusable Object-Oriented Software, Gamma, et al.