20.7.3   How do I keep persistent data across program calls?

For some specific applications, you can use one of the DBM modules. See AnyDBM_File. More generically, you should consult the FreezeThaw or Storable modules from CPAN. Starting from Perl 5.8 Storable is part of the standard distribution. Here's one example using Storable's `store` and `retrieve` functions:

```
use Storable;
store(\%hash, "filename");

# later on...
$href = retrieve("filename");        # by ref
%hash = %{ retrieve("filename") };   # direct to hash
```

20.7.4   How do I print out or copy a recursive data structure?

The Data::Dumper module on CPAN (or the 5.005 release of Perl) is great for printing out data structures. The Storable module on CPAN (or the 5.8 release of Perl), provides a function called `dclone` that recursively copies its argument.

```
use Storable qw(dclone);
$r2 = dclone($r1);
```

Where $r1 can be a reference to any kind of data structure you'd like. It will be deeply copied. Because `dclone` takes and returns references, you'd have to add extra punctuation if you had a hash of arrays that you wanted to copy.

```
%newhash = %{ dclone(\%oldhash) };
```

20.7.5   How do I define methods for every class/object?

Use the UNIVERSAL class (see *UNIVERSAL*).

20.7.6   How do I verify a credit card checksum?

Get the Business::CreditCard module from CPAN.

20.7.7   How do I pack arrays of doubles or floats for XS code?

The kgbpack.c code in the PGPLOT module on CPAN does just this. If you're doing a lot of float or double processing, consider using the PDL module from CPAN instead–it makes number-crunching easy.

20.8   AUTHOR AND COPYRIGHT