# Table of Contents

Foreword ................................................................. ix  
Preface ................................................................. xi  

1. **Introduction** .......................................................... 1  
   What Should You Know Already? 2  
   What About All Those Footnotes? 2  
   What's with the Exercises? 2  
   What if I'm a Perl Course Instructor? 3

2. **Building Larger Programs** ....................................... 4  
   The Cure for the Common Code 4  
   Inserting Code with eval 5  
   Using do 6  
   Using require 8  
   require and @INC 9  
   The Problem of Namespace Collisions 11  
   Packages as Namespace Separators 12  
   Scope of a Package Directive 14  
   Packages and Lexicals 15  
   Exercises 15

3. **Introduction to References** .................................... 17  
   Performing the Same Task on Many Arrays 17  
   Taking a Reference to an Array 19  
   Dereferencing the Array Reference 20  
   Dropping Those Braces 21  
   Modifying the Array 22
Nested Data Structures 23
Simplifying Nested Element References with Arrows 25
References to Hashes 26
Exercises 28

4. References and Scoping .................................. 30
   More than One Reference to Data 30
   What if That Was the Name? 31
   Reference Counting and Nested Data Structures 32
   When Reference Counting Goes Bad 34
   Creating an Anonymous Array Directly 35
   Creating an Anonymous Hash 38
   Autovivification 40
   Autovivification and Hashes 42
   Exercises 44

5. Manipulating Complex Data Structures .................. 45
   Using the Debugger to View Complex Data 45
   Viewing Complex Data with Data::Dumper 50
   Storing Complex Data with Storable 51
   The map and grep Operators 53
   Using map 55
   Applying a Bit of Indirection 56
   Selecting and Altering Complex Data 58
   Exercises 59

6. Subroutine References .................................. 61
   Referencing a Named Subroutine 61
   Anonymous Subroutines 65
   Callbacks 67
   Closures 68
   Returning a Subroutine from a Subroutine 70
   Closure Variables as Inputs 73
   Closure Variables as Static Local Variables 73
   Exercise 75

7. Practical Reference Tricks .............................. 77
   Review of Sorting 77
   Sorting with Indices 78
## 10. Object Destruction
- Beating a Dead Horse
- Indirect Object Notation
- Additional Instance Variables in Subclasses
- Using Class Variables
- Weakening the Argument
- Exercise

## 11. Some Advanced Object Topics
- UNIVERSAL Methods
- Testing Your Objects for Good Behavior
- AUTOLOAD as a Last Resort
- Using AUTOLOAD for Accessors
- Creating Getters and Setters More Easily
- Multiple Inheritance
- References to Filehandles
- Exercise

## 12. Using Modules
- Sample Function-Oriented Interface: File::Basename
- Selecting What to Import
- Sample Object-Oriented Interface: File::Spec
- A More Typical Object-Oriented Module: Math::BigInt
- The Differences Between OO and Non-OO Modules
- What use Is Doing
- Setting the Path at the Right Time
- Importing with Exporter
- @EXPORT and @EXPORT_OK
- Exporting in a Primarily OO Module
- Custom Import Routines
- Exercise

## 13. Writing a Distribution
- Starting with h2xs
- Looking at the Templates
- The Prototype Module Itself
- Embedded Documentation
- Controlling the Distribution with Makefile.PL
- Alternate Installation Locations (PREFIX=...)

---

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>10. Object Destruction</td>
<td>112</td>
</tr>
<tr>
<td>Beating a Dead Horse</td>
<td>117</td>
</tr>
<tr>
<td>Indirect Object Notation</td>
<td>118</td>
</tr>
<tr>
<td>Additional Instance Variables in Subclasses</td>
<td>119</td>
</tr>
<tr>
<td>Using Class Variables</td>
<td>121</td>
</tr>
<tr>
<td>Weakening the Argument</td>
<td>122</td>
</tr>
<tr>
<td>Exercise</td>
<td>125</td>
</tr>
<tr>
<td>11. Some Advanced Object Topics</td>
<td>126</td>
</tr>
<tr>
<td>UNIVERSAL Methods</td>
<td>126</td>
</tr>
<tr>
<td>Testing Your Objects for Good Behavior</td>
<td>127</td>
</tr>
<tr>
<td>AUTOLOAD as a Last Resort</td>
<td>128</td>
</tr>
<tr>
<td>Using AUTOLOAD for Accessors</td>
<td>129</td>
</tr>
<tr>
<td>Creating Getters and Setters More Easily</td>
<td>130</td>
</tr>
<tr>
<td>Multiple Inheritance</td>
<td>132</td>
</tr>
<tr>
<td>References to Filehandles</td>
<td>133</td>
</tr>
<tr>
<td>Exercise</td>
<td>135</td>
</tr>
<tr>
<td>12. Using Modules</td>
<td>137</td>
</tr>
<tr>
<td>Sample Function-Oriented Interface: File::Basename</td>
<td>137</td>
</tr>
<tr>
<td>Selecting What to Import</td>
<td>138</td>
</tr>
<tr>
<td>Sample Object-Oriented Interface: File::Spec</td>
<td>138</td>
</tr>
<tr>
<td>A More Typical Object-Oriented Module: Math::BigInt</td>
<td>139</td>
</tr>
<tr>
<td>The Differences Between OO and Non-OO Modules</td>
<td>140</td>
</tr>
<tr>
<td>What use Is Doing</td>
<td>140</td>
</tr>
<tr>
<td>Setting the Path at the Right Time</td>
<td>141</td>
</tr>
<tr>
<td>Importing with Exporter</td>
<td>143</td>
</tr>
<tr>
<td>@EXPORT and @EXPORT_OK</td>
<td>144</td>
</tr>
<tr>
<td>Exporting in a Primarily OO Module</td>
<td>145</td>
</tr>
<tr>
<td>Custom Import Routines</td>
<td>147</td>
</tr>
<tr>
<td>Exercise</td>
<td>147</td>
</tr>
<tr>
<td>13. Writing a Distribution</td>
<td>148</td>
</tr>
<tr>
<td>Starting with h2xs</td>
<td>149</td>
</tr>
<tr>
<td>Looking at the Templates</td>
<td>149</td>
</tr>
<tr>
<td>The Prototype Module Itself</td>
<td>152</td>
</tr>
<tr>
<td>Embedded Documentation</td>
<td>154</td>
</tr>
<tr>
<td>Controlling the Distribution with Makefile.PL</td>
<td>158</td>
</tr>
<tr>
<td>Alternate Installation Locations (PREFIX=...)</td>
<td>159</td>
</tr>
</tbody>
</table>