

# Table of Contents

<b>Table of Contents .....</b>	<b>i</b>
<b>Module 1: Course Introduction and &lt;cfchart&gt;.....</b>	<b>1</b>
Introduction.....	1
Course Objectives .....	2
Course Prerequisites .....	2
Course Project.....	2
Exercise 1: Set-up Course Files and Data Sources (DSNs) .....	5
<b>Charting and Graphing Options .....</b>	<b>7</b>
Demo: Basic Charts .....	7
Using <cfchart> vs. <cfchartseries> .....	9
Demo: Linking from a chart to another page .....	11
Demo: <cfchartdata> to manually add data .....	13
Caching Charts.....	15
Demo: Save charts for future use with <cffile> .....	15
Exercise 2: Create a chart with <cfchart> .....	17
Possible Solution to Exercise 2 .....	19
<b>Module 2: Manipulating Complex Data Objects (Lists, Arrays, Structures and Queries).....</b>	<b>21</b>
Scalar Variables .....	21
Lists.....	22
Arrays .....	22
Structures .....	23
Exercise 3: Arrays and Lists .....	25
Possible Solution to Exercise 3 .....	27
<b>Looping with &lt;cfloop&gt;.....</b>	<b>29</b>
Syntax of <cfloop> .....	32
Weeding Out the FIELDNAMES Variable .....	32
Using <cfloop> for automatic query output.....	33
Display Web-safe Colors using CFLOOP .....	35
Exercise 4: Display results with <cfloop> .....	37
Possible Solution to Exercise 4 .....	39
<b>Queries of Queries.....</b>	<b>41</b>
Demo: Query of Query .....	41
Exercise 5: Query of query to display “classic movies” .....	45
Possible Solution to Exercise 5 .....	47
<b>Module 3: CFCs - Building ColdFusion Components.....</b>	<b>49</b>
Benefits to using CFCs .....	49

Creating Components.....	50
Access control.....	51
Calling or Invoking Components.....	52
Passing Arguments to Components .....	53
The main CFML page.....	56
Three Ways to Pass Arguments .....	57
Where to save CFCs .....	58
Self-documenting - View a Component directly in the browser.....	58
Exercise 6: Creating ColdFusion Components with Arguments.....	61
A Possible Solution to Exercise 6.....	64
<b>Object-Oriented CFCs.....</b>	<b>67</b>
Object-Oriented Principles.....	67
Demo: CFC using Object-Oriented Model .....	68
How does use of a CFC compare to other OO languages? .....	71
Demo: CFC using Object-Oriented Model .....	72
Inheritance .....	76
Demo: Inherit from another CFC.....	76
Exposing CFCs as Web Services.....	77
Exercise 7: Build OOP CFC with Parameters to Edit Courses .....	79
A Possible Solution to Exercise 7.....	83
<b>Module 4: Application Framework.....</b>	<b>85</b>
Automatic Includes: Application.cfm and onRequestEnd.cfm .....	85
New Application Framework: Application.cfc .....	87
Exercise 8: Store DSN and Company Name as Application Variables .....	91
Possible Solution to Exercise 8.....	93
Using Application Variables.....	95
Exercise 9: Cycle through Featured Movies Using Application Variables.....	97
Possible Solution to Exercise 9.....	99
<b>Module 5: Errors and Exceptions .....</b>	<b>101</b>
<b>Standard Error Handling.....</b>	<b>102</b>
Application.cfm .....	103
Demo: the error page .....	103
Demo: detailed error page.....	104
Exercise 10: Replace the Default Error Page.....	107
Possible Solution to Exercise 10.....	109
<b>Structured Error Handling.....</b>	<b>111</b>
<cftry> and <cfcatch>.....	111
Demo: Using <cftry> .....	111
<b>Transactions with &lt;cftransaction&gt; and &lt;cftry&gt; .....</b>	<b>113</b>
Classic <cftransaction> example .....	113
Demo: Using <cftransaction> with <cftry> .....	113

Exercise 11: Using <cftry> to Avoid Errors .....	115
Possible Solution to Exercise 11 .....	117
<b>Module 6: Reusing Code .....</b>	<b>119</b>
Limited Capabilities of <cfscript> .....	119
Syntax Rules .....	120
Reserved Words .....	120
Looping with <cfscript> .....	120
GetTickCount() - testing the speed of your code .....	122
Exercise 12: Setting Variables with <cfscript> .....	125
Possible Solution to Exercise 12 .....	127
<b>Custom Tags .....</b>	<b>129</b>
Demo: custom tags .....	129
Where to store Custom Tags .....	132
Demo: Custom Tag with Parameters .....	133
CALLER and ATTRIBUTE scopes .....	135
Using <cfmodule> .....	136
Exercise 13: Using Custom Tags .....	137
Possible Solution to Exercise 13 .....	139
<b>Stored Procedures .....</b>	<b>141</b>
Benefits of Stored Procedures .....	141
Demo: Stored Procedure .....	141
Demo: pass a parameter to a stored procedure .....	143
Exercise 14: Pass Parameters to a Stored Procedure .....	145
Possible Solution to Exercise 14 .....	147
<b>Showing portions of a Recordset .....</b>	<b>149</b>
Paging through records .....	150
Exercise 15: Add “Next” and “Previous” links to the Actors list .....	155
Possible Solution to Exercise 15 .....	157
<b>Module 7: Advanced Validation with Regular Expressions (optional) .....</b>	<b>159</b>
Special Symbols for Matching Specific Characters .....	159
Specify the number of occurrences .....	160
Placement of the pattern .....	160
POSIX Syntax .....	161
Perl Syntax .....	161
Demo: Testing Regular Expressions .....	161
Exercise 16: Using Regular Expressions for Advanced Form Validation .....	165
Possible Solution to Exercise 16 .....	169
<b>Appendix A: Resources .....</b>	<b>171</b>
<b>Appendix B: Set-up Notes .....</b>	<b>173</b>
<b>Appendix C: Selecting Among ColdFusion Code Reuse Methods.....</b>	<b>175</b>