

Contents (preliminary, June 2006)

1	The Integrated Development Environment.....	1
1.1	Visual Programming: A First Small Program.....	2
1.2	First steps in C++.....	5
1.3	The Code Editor.....	7
1.4	Context Menus.....	11
1.5	Projects, Project Files and Project Options.....	11
1.6	Some Tips when Working with Projects.....	14
1.7	The Online Help.....	17
1.8	Toolbars.....	19
1.9	Project Groups and the Project Manager Θ.....	20
1.10	Tools for Designing Forms Θ.....	21
1.11	Packages and Standalone Executable Programs Θ.....	23
1.12	Win32-API and Console Applications Θ.....	24
1.12.1	Console Applications Θ.....	24
1.12.2	Starting the Compiler from a Command Prompt Θ.....	25
1.12.3	Win32 API Applications Θ.....	26
1.13	Windows Programs and Units Θ.....	26
2	User Interface Controls.....	31
2.1	The Online Help for the Components.....	32
2.2	Names.....	35
2.3	Labels, Data Types and Compiler Error Messages.....	38
2.4	Functions, Methods and the <i>TEdit</i> Control.....	43
2.5	Memos, ListBoxes, ComboBoxes and the <i>TStrings</i> Class.....	47
2.6	Buttons and Events.....	51
2.7	CheckBoxes, RadioButtons and Simple <i>if</i> Statements.....	56
2.8	The GroupBox, Panel and RadioGroup Containers.....	58
2.9	Main Menu and Context Menus.....	60
2.9.1	Main Menus and the Menu Designer.....	61
2.9.2	Context Menus.....	62

2.9.3	Managing Images: ImageList Θ	63
2.9.4	Saving and Loading Menu Templates Θ	64
2.10	Common Dialog Boxes.....	65

3 Fundamental Types and Statements 68

3.1	Syntax Rules.....	68
3.2	Variables and Identifiers.....	72
3.3	Integral Data Types.....	78
3.3.1	The Internal Representation of Integral Values.....	81
3.3.2	Integer Literals and their Type.....	84
3.3.3	Assignments and Standard Conversions with Integral Expressions.....	86
3.3.4	Operators and the “Usual Arithmetic Conversions”.....	90
3.3.5	The <i>bool</i> Type.....	97
3.3.6	The <i>char</i> Data Types and the ASCII and ANSI Character Sets ..	101
3.3.7	The <i>__int64</i> Type.....	109
3.4	Control Structures and Functions.....	110
3.4.1	The <i>if</i> and the Compound Statement.....	110
3.4.2	Iteration Statements.....	115
3.4.3	Functions and the Data Type <i>void</i>	120
3.4.4	Value Parameters and Reference Parameters.....	125
3.4.5	Using Libraries.....	125
3.4.6	Random Numbers.....	126
3.5	Tests and the Integrated Debugger.....	131
3.5.1	Systematic Testing.....	131
3.5.2	Test Protocols and Automatic Test functions.....	136
3.5.3	The Integrated Debugger.....	140
3.6	Floating-Point Types.....	148
3.6.1	The Internal Encoding of Floating-Point Values.....	149
3.6.2	The Type of Floating-Point Literals.....	153
3.6.3	Standard Conversions.....	155
3.6.4	Mathematical Functions.....	162
3.6.5	Types for Exact and Financial Calculations.....	165
3.6.6	A Test for Nearly Equal Floating-Point Numbers.....	169
3.7	Trace Tables and Programming Logic.....	176
3.7.1	Trace Tables.....	177
3.7.2	Symbolic Trace Tables.....	184
3.7.3	Loop Invariants Θ	190
3.7.4	Program Verification, Tests and Checking Conditions at Runtime.....	197
3.7.5	Function Calls and Programming Style for Functions.....	201
3.7.6	Simple Rules of Logic and Truth Tables Θ	206
3.7.7	Conditions in the Branches of <i>if</i> Statements and Loops.....	208
3.8	Constants.....	216
3.9	Syntax Rules for Simple Declarations and Initializations Θ	220
3.10	Pointers, Strings and Dynamically Created Variables.....	221
3.10.1	The Definition of Pointer Variables.....	223

3.10.2	The Address Operator, Assignments and Generic Pointers	225
3.10.3	Trace Tables for Pointer Variables	229
3.10.4	Dynamically Created Variables: <i>new</i> and <i>delete</i>	231
3.10.5	Pointer Arithmetic and Arrays	240
3.10.6	<i>const</i> Pointers	241
3.10.7	String Literals, Null-terminated Strings and <i>char*</i> Pointers.....	244
3.10.8	Pointers as Parameters and Windows API Pointer Types.....	248
3.10.9	Library Functions for Null-Terminated Strings	255
3.10.10	Pointers to Pointers to Pointers to	259
3.10.11	Reference Types and Reference Parameters Θ	260
3.11	Declarations with <i>typedef</i> and <i>typeid</i> Expressions	263
3.12	Enumeration Types	266
3.12.1	<i>enum</i> Constants and Conversions	270
3.13	Comments and Internal Program Documentation.....	273
3.14	Global, Local and Dynamic Variables.....	277
3.14.1	The Declaration Statement.....	277
3.14.2	The Compound Statement and Local Scope	278
3.14.3	Local <i>static</i> Variables	281
3.14.4	Lifetime of Variables and Storage Class Specifiers Θ	282
3.15	Preprocessor Directives	285
3.15.1	The <i>#include</i> directive.....	286
3.15.2	Macros	288
3.15.3	Conditional Compilation.....	293
3.15.4	Pragmas.....	301
3.16	More Statements	304
3.16.1	The Expression Statement.....	305
3.16.2	Exception Handling: <i>try</i> and <i>throw</i>	307
3.16.3	The <i>switch</i> Statement	314
3.16.4	The <i>do</i> Statement	316
3.16.5	The <i>for</i> Statement.....	317
3.16.6	The Jump Statements <i>goto</i> , <i>break</i> and <i>continue</i>	320
3.17	Expressions	323
3.17.1	Primary Expressions Θ	324
3.17.2	Postfix Expressions Θ	326
3.17.3	Unary expressions Θ	327
3.17.4	Type Conversions in Cast Notation Θ	331
3.17.5	Pointers to Class Members Θ	331
3.17.6	Multiplicative Operators Θ	331
3.17.7	Additive Operators Θ	331
3.17.8	Shift Operators Θ	332
3.17.9	Relational Operators Θ	333
3.17.10	Equality Operators Θ	333
3.17.11	Bitwise Operators Θ	334
3.17.12	Logical Operators Θ	335
3.17.13	The Conditional Operator Θ	336
3.17.14	Constant Expressions Θ	337
3.17.15	Assignment Operators Θ	338

3.17.16	The Comma Operator Θ	340
3.17.17	Lvalues and Rvalues Θ	341
3.17.18	The Precedence and Associativity of the Operators Θ	341
3.17.19	Alternative Tokens Θ	343
3.17.20	Explicit Type Conversions Θ	346
4	Structured Data Types and Predefined Classes.....	355
4.1	The String Classes <i>AnsiString</i> and <i>string</i>	355
4.1.1	Using Member Functions of Predefined Classes.....	357
4.1.2	Similarities and Differences of the String Classes.....	359
4.1.3	Functions for the Class <i>AnsiString</i>	363
4.1.4	Some Member Functions of the Class <i>string</i>	367
4.2	Arrays and Containers.....	374
4.2.1	One-dimensional Arrays.....	374
4.2.2	Initializing Arrays in their Definition.....	382
4.2.3	Dynamically created one dimensional arrays.....	384
4.2.4	Arrays as Containers.....	386
4.2.5	Multidimensional Arrays.....	392
4.2.6	Arrays, Pointers, and Pointer Arithmetic.....	398
4.2.7	Arrays as Function Parameters Θ	402
4.2.8	Dynamically Created Multidimensional Arrays Θ	406
4.2.9	Array Properties from the VCL.....	410
4.3	Standard Library Sequential Containers.....	411
4.3.1	The Container Class <i>vector</i>	412
4.3.2	Iterators.....	415
4.3.3	Standard Library Algorithms.....	419
4.3.4	Vector Memory Management.....	422
4.3.5	Multidimensional Vectors.....	423
4.3.6	The Container Classes <i>list</i> and <i>deque</i>	427
4.3.7	Common Features and Differences of the Sequential Containers.....	429
4.3.8	The Container Adapters <i>stack</i> , <i>queue</i> and <i>priority_queue</i>	431
4.3.9	Containers with Pointers.....	432
4.4	Structures and Classes.....	437
4.4.1	Classes Defined with <i>struct</i>	437
4.4.2	Linked Lists.....	447
4.4.3	Memory Leaks and Heap Control Θ	457
4.4.4	Data structures with generic pointers Θ	459
4.4.5	Classes Defined with <i>union</i> Θ	460
4.4.6	The Data Types <i>TVarRec</i> and <i>Variant</i> Θ	463
4.4.7	Bit-Fields Θ	465
4.5	Files.....	467
4.5.1	Stream Variables, their Connection to Files, and their State.....	467
4.5.2	Errors and the State of Stream Variables.....	471
4.5.3	Reading and Writing Binary Data with <i>read</i> and <i>write</i>	473
4.5.4	Reading and Writing Data with the << and >> Operators.....	480

4.5.5	Manipulators and Text Formatting Functions.....	489
4.5.6	String Streams.....	491
4.5.7	Random Access Files.....	493
4.5.8	Sorting, Merging and Group Processing Θ	496
4.5.9	C File Processing Functions Θ	503
4.6	Associative Containers.....	506
4.6.1	The <i>set</i> and <i>multiset</i> Containers.....	506
4.6.2	The <i>map</i> and <i>multimap</i> Containers.....	507
4.6.3	Associative Containers Iterators.....	509
4.6.4	Hash Containers.....	510
4.7	The Standard Library Numeric Classes.....	512
4.7.1	Complex Numbers.....	512
4.7.2	Valarrays and Slices Θ	516

5 Functions 537

5.1	The Definition and Calling of Functions.....	538
5.2	The Implementation of Function Calls on the Stack.....	541
5.3	Functions with Parameters.....	544
5.3.1	Value Parameters.....	546
5.3.2	Call by Reference with Reference Parameters.....	548
5.3.3	Pointer Parameters.....	550
5.3.4	Constant Parameters.....	553
5.3.5	Side Effects and the Order of Evaluation.....	555
5.3.6	Default Arguments.....	560
5.3.7	The Data Type of a Function.....	562
5.3.8	Pointers to Functions.....	563
5.3.9	Unspecified Number and Types of Arguments Θ	572
5.3.10	The Functions <i>main</i> and <i>WinMain</i> and their Parameters.....	573
5.3.11	Calling Delphi Functions in C++Builder Θ	575
5.3.12	Traditional K&R Function Definitions Θ	577
5.3.13	Calling Conventions.....	578
5.4	Stepwise Refinement as a Design Technique.....	578
5.6	Recursion.....	582
5.6.1	Basic Principles of Recursion.....	583
5.6.2	Quicksort.....	587
5.6.3	A Recursive Descent Parser.....	591
5.6.4	Recursively Defined Curves.....	594
5.6.5	Indirect Recursion.....	596
5.6.6	Recursive Data Structures and Binary Search Trees.....	600
5.6.7	Recursive Data Structures in the C++ Standard Library.....	605
5.7	Inline Functions.....	608
5.8	Overloaded Functions.....	609
5.9	Overloaded Operators with Global Operator Functions.....	618
5.9.1	Global Operator Functions.....	619

5.9.2	The Increment and Decrement Operators	621
5.9.3	Reference Return Types.....	623
5.9.4	The Input and Output of User-Defined Data Types.....	625
6	Object-Oriented Programming	627
6.1	Classes	628
6.1.1	Data Members and Member Functions	628
6.1.2	The Scope of Class Members	633
6.1.3	Objects and Member Access Control: <i>private</i> and <i>public</i>	635
6.1.4	Calling Member Functions and the <i>this</i> Pointer.....	641
6.1.5	Constructors and Destructors	643
6.1.6	OO analysis and design: The design of classes.....	656
6.1.7	A little programming logic: Class invariants and correctness.....	660
6.1.8	UML diagrams for classes and objects	665
6.2	Classes as Data Types.....	668
6.2.1	The Default Constructor	669
6.2.2	Objects as Class Members and Member Initializers	671
6.2.3	<i>Friend</i> Functions and Classes	677
6.2.4	Overloaded Operators as Member Functions.....	680
6.2.5	The Copy Constructor.....	688
6.2.6	The Assignment Operator = for Classes	695
6.2.7	User-Defined Conversions.....	703
6.2.8	Explicit Constructors Θ	707
6.2.9	Static Class Members.....	709
6.2.10	Constant Class Members and Objects.....	711
6.2.11	Further Declaration Options in Classes Θ	713
6.2.12	Classes and Header Files	714
6.2.13	Overloaded Operators for <i>new</i> and <i>delete</i> Θ	716
6.3	Inheritance	718
6.3.1	The Members of Derived Classes	719
6.3.2	Access Rights to the Members of Base Classes	721
6.3.3	The Meaning of Member Names in a Class Hierarchy	723
6.3.4	<i>Using</i> Declarations in Derived Classes Θ	725
6.3.5	Constructors, Destructors and Implicitly Created Functions	726
6.3.6	Inheritance for Forms in the C++ Builder.....	732
6.3.7	OO Design: <i>public</i> Inheritance and “is-a” Relations	733
6.3.8	OO Design: Composition and “has-a” Relations	738
6.3.9	Conversions Between <i>Public</i> Derived Classes	740
6.3.10	<i>Protected</i> and <i>Private</i> Derived Classes.....	743
6.3.11	Multiple Inheritance and Virtual Base Classes	747
6.4	Virtual Functions, Late Binding and Polymorphism	753
6.4.1	The Static and the Dynamic Data Type	753
6.4.2	Virtual Functions	754
6.4.3	The Internal Implementation of Virtual Functions: <i>vptr</i> and <i>vtbl</i>	765
6.4.4	OO Design: The Application Area of Virtual Functions	770

6.4.5	Composition and <i>private</i> Multiple Inheritance Θ	772
6.4.6	Virtual Constructors and Destructors.....	773
6.4.7	Virtual Functions in Constructors and Destructors Θ	775
6.4.8	Virtual Functions and Extensibility	776
6.4.9	Purely Virtual Functions and Abstract Classes.....	779
6.4.10	OO Design: Virtual Functions and Abstract Base Classes	782
6.4.11	Protocol Classes and Application Frameworks	786
6.4.12	Scrutinize (Patterns).....	788
6.4.13	UML Diagrams for Inheritance and Composition	793
6.4.14	Pointers to Class Members Θ	795
6.5	Run-Time Type Information.....	799
6.5.1	Type Information with the Operator <i>typeid</i> Θ	800
6.5.2	Type Conversions with <i>dynamic_cast</i> Θ	803
6.5.3	Use of Run-Time Information Θ	807
6.5.4	<i>static_cast</i> with Classes Θ	809
6.5.5	Run-Time Type Information for VCL Classes Θ	812
7	Exception Handling	817
7.1	The <i>try</i> statement	819
7.2	Exception Handlers and Standard Library Exceptions	822
7.3	Predefined Exceptions in the VCL	826
7.4	Program Flow on Exceptions.....	829
7.5	Predefined Exception Handling in the VCL	831
7.6	<i>throw</i> Expressions and self defined exceptions	832
7.7	Errors, Exceptions and Program Correctness	838
7.8	Releasing Resources on Exceptions	841
7.9	Exceptions in <math.h> and the Function <i>_matherr</i> Θ	847
7.10	The Class <i>auto_ptr</i> Θ	848
7.11	Exception Specifications.....	850
7.12	The Function <i>terminate</i> Θ	852
7.13	Win32 Exception Handling with <i>try-__except</i> Θ	852
8	The Visual Component Library (VCL)	855
8.1	Special Features of the VCL.....	856
8.2	Visual Programming and Properties	860
8.2.1	Reading and Writing Properties.....	860
8.2.2	Array Properties Θ	863
8.2.3	Index Specifiers	864
8.2.4	Specifiers for Storing Properties in the Form File	865
8.2.5	Redeclaring Properties	867
8.3	The VCL Class Hierarchy.....	867
8.4	Custom Components and their Events.....	875
8.5	Extending the Component Palette.....	882

8.7	Class Reference Types and Virtual Constructors	894
8.8	Messages	898
8.8.1	The Message Queue and the Window Procedure	898
8.8.2	Messages for an Application	901
8.8.3	Messages for a Control and Message Handling in the VCL Classes	902
8.8.4	Custom Message Handlers	904
8.8.5	Sending Messages	909

9 Templates and the STL 914

9.1	Generic Functions: Function Templates	915
9.1.1	The Declaration of Function Templates with Type Parameters	916
9.1.2	Specializations of Function Templates	918
9.1.3	Function Templates with Non-Type Parameters	924
9.1.4	Explicitly instantiated function templates \oplus	926
9.1.5	Explicitly specialized and overloaded templates	927
9.1.6	Recursive function templates	930
9.2	Generic Classes: Class templates	933
9.2.1	The declaration of class templates with type parameters	933
9.2.2	Specializations of class templates	934
9.2.3	Templates with non-type parameters Θ	942
9.2.4	Explicitly instantiated class templates Θ	943
9.2.5	Partial and complete specialization Θ	944
9.2.6	Metaprogramming and Type Traits	950
9.2.7	Members and friend functions of class templates Θ	953
9.2.8	Template Derivations Θ	956
9.2.9	Exported templates	957
9.2.10	UML-Diagrams for templates Θ	958
9.3	Funktionsobjekte in der STL	962
9.3.1	The Call Operator ()	962
9.3.2	Predicates and Arithmetic Function Objects	965
9.3.3	Binder, Funktionsadapter und Negatoren	968
9.3.4	C++0X Extensions concerning Function Objects	973
9.4	Iterators and the STL Algorithms	977
9.4.1	The Different Kinds of Iterators	978
9.4.2	Reverse Iterators	980
9.4.3	Insert Functions and Insert Iterators	981
9.4.4	Stream Iterators	983
9.4.5	Container Constructors with Iterators	984
9.4.6	STL Algorithms for all Elements of a Container	985
9.5	Die Algorithmen der STL	987
9.5.1	Lineares Suchen	988
9.5.2	Zählen	990
9.5.3	Der Vergleich von Bereichen	990
9.5.4	Suche nach Teilfolgen	992
9.5.5	Minimum und Maximum	993

9.5.6	Elemente vertauschen	995
9.5.7	Kopieren von Bereichen	995
9.5.8	Elemente transformieren und ersetzen	997
9.5.9	Elementen in einem Bereich Werte zuweisen	999
9.5.10	Elemente entfernen	999
9.5.11	Die Reihenfolge von Elementen vertauschen	1001
9.5.12	Permutationen	1002
9.5.13	Partitionen	1003
9.5.14	Bereiche sortieren	1003
9.5.15	Binäres Suchen in sortierten Bereichen	1005
9.5.16	Mischen von sortierten Bereichen	1006
9.5.17	Mengenoperationen auf sortierten Bereichen	1008
9.5.18	Heap-Operationen	1009
9.5.19	Verallgemeinerte numerische Operationen	1010
10	Miscellaneous	1013
10.1	Toolbars, Menus and Actions	1014
10.1.1	Toolbars made of Panels and SpeedButtons	1014
10.1.2	ToolBar	1014
10.1.3	Hosts for Moveable Controls: CoolBar and ControlBar	1015
10.1.4	Managing Actions	1015
10.2	Custom Dialogs, Frames and the Object Repository	1021
10.2.1	Displaying Additional Forms and Modal Forms	1021
10.2.2	Dialog Boxes from the Object Repository	1024
10.2.3	Functions that display Predefined Dialog Boxes	1024
10.2.4	Frames - Installing Custom Components on the Tool-Palette ...	1026
10.2.5	Data Modules	1027
10.2.6	The Object Repository	1028
10.3	ListView and TreeView	1028
10.3.1	Displaying Lists with a ListView	1028
10.3.2	Displaying Trees with TreeView	1030
10.3.3	Components for resizing Components	1033
10.4	Formatted Texts with the RichEdit Component	1036
10.5	Time-controlled Events with a Timer	1037
10.6	Tables	1038
10.7	Slider Controls: ScrollBar and TrackBar	1040
10.8	More Input Controls	1042
10.8.1	Filtering Text Input	1042
10.8.2	Selecting Drives and Directories	1044
10.8.3	Other Controls	1045
10.9	More Output Controls	1046
10.9.1	Status information	1046
10.9.2	Progression indicated in progress bar	1047
10.9.3	Multimedia Components	1048
10.10	FlowPanel and GridPanel	1049

10.11	TrayIcon.....	1052
10.14	Images and Graphics.....	1058
10.14.1	Components to display Graphics	1058
10.14.2	Graphics	1060
10.14.3	Printing with <i>TPrinter</i> and <i>TRichEdit</i>	1065
10.14.4	TChart.....	1067
10.15	Automating MS Office Applications: Creating Word-Documents...1074	
10.16	Internet-Components	1077
10.17	MDI Programs	1082
10.19	Monetary Values: The Classes <i>Currency</i> and <i>BCD</i>	1085
10.20	Classes and Functions for Date and Time	1086
10.20.1	High Resolution Timer	1088
10.20.2	Entering Times and Dates	1089
10.21	The Class <i>Set</i>	1092
10.22	3D Graphics with OpenGL.....	1095
10.22.1	Initializations.....	1095
10.22.2	Basic Elements of Graphics: Primitives.....	1098
10.22.3	Modeling Transformations.....	1103
10.22.4	Predefined Bodies	1106
10.22.5	Local Transformations	1107
10.22.6	Lights Effects.....	1110
10.22.7	Textures	1113
10.23	Win32 File Processing Functions	1117
10.23.1	Basic Functions.....	1117
10.23.2	Simultaneous Access of Several Users to a File	1120
10.23.3	Record Locking.....	1121
10.23.4	VCL File Processing Functions and the Class <i>TFileStream</i> ..1122	
10.24	Serial Port Data Transfer	1124
10.24.1	Basic Concepts.....	1125
10.24.2	Serial Interface Standards: RS 232 C or V.24	1126
10.24.3	Win32 Functions for Serial Communication	1128
10.25	ADO Database Components.....	1131
10.25.1	Tables and the TTable Component	1132
10.25.2	Reading and Writing Table Data	1135
10.25.3	Displaying Tables with the DBGrid Component.....	1138
10.25.6	SQL Queries	1146