

**OUTLINE**  
**DELPHI JUMP START**  
**OBJECT-ORIENTED PROGRAMMING AND APPLICATION DEVELOPMENT**  
**IN DELPHI**

**CARY JENSEN, PH.D.**

COPYRIGHT 1995-2004. CARY JENSEN. JENSEN DATA SYSTEMS, INC.

ALL RIGHTS RESERVED.

JENSEN DATA SYSTEMS, INC.

HTTP://WWW.JENSEN DATASYSTEMS.COM

PHONE: 281-359-3311

EMAIL: INFO@JENSEN DATASYSTEMS.COM

**DELPHI JUMP START: OBJECT-ORIENTED PROGRAMMING AND APPLICATION DEVELOPMENT IN DELPHI .....**

**DELPHI OVERVIEW .....**

RAD: BUILDING AN APPLICATION NOW .....

**UNDERSTANDING COMPONENTS .....**

BASIC COMPONENT FEATURES .....

GETTING HELP ABOUT A COMPONENT.....

**THE OBJECT REPOSITORY .....**

WIZARDS .....

TEMPLATES .....

*Saving a Form Template.....*

*Saving a Project Template.....*

VISUAL FORM INHERITANCE .....

USING A TEMPLATE EXAMPLE.....

FILES OF AN APPLICATION .....

**BUILDING DATABASE APPLICATIONS IN DELPHI.....**

EXPLORING DATA ACCESS COMPONENTS .....

*Connections and DataSets.....*

*The Connection Component.....*

*DataSets.....*

USING THE OBJECT INSPECTOR .....

**EVENT HANDLERS .....**

**ADDING MENUS AND ADDITIONAL FORMS .....**

CREATING MENUS .....

ADDING MENU CODE.....

CREATING POPUP MENUS .....

ADDING ADDITIONAL FORMS.....

ACCESSING AN ADDED FORM.....

CREATING OBJECTS AT RUNTIME AND GARBAGE COLLECTION .....

CREATING AND RELEASING NONMODAL FORMS .....

**WORKING WITH FIELDS.....**

- USING INSTANTIATED FIELDS.....
- USING THE FIELDS PROPERTY .....
- USING THE FIELDBYNAME METHOD .....
- CONTROLLING TAB ORDER.....**
- EXCEPTION HANDLING IN DELPHI.....**
  - WHAT HAPPENS WHEN AN EXCEPTION OCCURS .....
  - PROTECTING AGAINST EXCEPTIONS .....
  - USING TRY ... FINALLY .....
  - USING TRY ... EXCEPT.....
  - MORE ABOUT EXCEPTIONS .....
  - CREATING YOUR OWN EXCEPTION HANDLERS .....
  - SILENT EXCEPTIONS .....
  - WHY RAISE AN EXCEPTION.....
- DATA VALIDATION .....**
  - FIELD-LEVEL VALIDATION.....
  - RECORD-LEVEL VALIDATION .....
  - FORM-LEVEL VALIDATION.....
    - Form-Level Validation with Datasets* .....
  - APPLICATION-LEVEL EXCEPTION HANDLING .....
- USING DATA MODULES.....**
  - USING A DATA MODULE EXAMPLE.....
  - DATA MODULE LIMITATIONS .....
  - Data Modules and Reports* .....
  - Multiple Instance Forms*.....
  - Specialized Form Views*.....
- CREATING AND USING DLLS .....**
  - OVERVIEW OF DLLS .....
  - BUILDING A DLL EXAMPLE .....
  - More About the Exports Clause* .....
  - DLL Parameters*.....
  - USING DLLS IN APPLICATIONS .....
  - CREATING DLL IMPORT UNITS .....
  - CREATING DLLS CONTAINING FORMS.....
  - DLLS AND EXCEPTIONS .....
  - EXPLICITLY LOADING AND FREEING DLLS .....
  - Declaring Variables* .....
  - Loading The DLL* .....
  - Getting the Function or Procedure Address*.....
  - Releasing The DLL*.....
  - DLL INITIALIZATION AND EXIT.....
    - Defining Initialization Code* .....
    - Defining an Exit Procedure* .....
  - TAPPLICATION AND DLLS .....
- POWER DEBUGGING TECHNIQUES.....**
  - BEFORE YOU BEGIN: SETTING UP THE DEBUGGER .....
  - Enabling and Disabling the Integrated Debugger*.....
  - Instructing the Debugger to Ignore Particular Exceptions*.....
  - Configuring the Event Log*.....
  - USING BREAKPOINTS .....

- Source Breakpoints* .....
- Controlling the Debugger at Runtime* .....
- Address Breakpoints* .....
- Data Breakpoints* .....
- Module Load Breakpoints* .....
- USING PROGRAM RESET .....
- ASSERTIONS .....
- Using Assertions* .....
- Enabling Assertions* .....
- RUN TO RETURN .....
- OUTPUTDEBUGSTRING .....
- DEBUGGING WITH HOST APPLICATIONS AND PROJECT GROUPS .....
- Debugging DLLs Using the Run Parameters Dialog Box* .....
- Debugging DLLs Using a Project Group* .....
- LEVERAGING VISUAL FORM INHERITANCE** .....
- ADVANTAGES OF VFI .....
- INHERITED METHODS .....
- OVERRIDING INHERITED METHODS .....
- DERIVING FROM THE CURRENT PROJECT .....
- DEFINING A SHARED OBJECT REPOSITORY .....
- COMPONENTS TO AVOID .....
- USING COMPONENT TEMPLATES** .....
- CREATING A COMPONENT TEMPLATE .....
- CREATING COMPONENT TEMPLATES WITH EVENT HANDLERS .....
- DELETING COMPONENT TEMPLATES .....
- COMPONENT TEMPLATE GUIDELINES .....
- UNDERSTANDING FRAMES** .....
- CREATING A FRAME .....
- USING A FRAME .....
- FRAME CLASSES AND FRAME INSTANCES .....
- OVERRIDING CONTAINED COMPONENT PROPERTIES .....
- Reverting Properties to their Original Values* .....
- CONTAINED OBJECT EVENT HANDLERS .....
- OVERRIDING CONTAINED OBJECT EVENT HANDLERS .....
- FRAMES THAT SAVE RESOURCES .....
- SIMPLIFYING FRAME USAGE .....
- Adding a Frame to the Component palette* .....
- Using a Frame from the Component palette* .....
- Adding a Frame to the Object Repository* .....
- Using a Frame from the Object Repository* .....
- TURNING FRAMES INTO TRUE COMPONENTS .....
- OBJECTS, COMPONENTS, AND PACKAGES** .....
- OVERVIEW OF PACKAGES** .....
- DESIGN-TIME PACKAGES .....
- CREATING A NEW DESIGN-TIME PACKAGE .....
- INSTALLING AN EXISTING DESIGN-TIME PACKAGE .....
- RUNTIME PACKAGES .....
- WHEN TO USE RUNTIME PACKAGES .....
- Multiple Applications* .....
- Small Executables* .....

- Modular Applications* .....
- HOW RUNTIME PACKAGES AFFECT APPLICATION DEVELOPMENT .....
- CONFIGURING A PACKAGE .....
- LOADING AND UNLOADING A PACKAGE AT RUNTIME .....
- DEPLOYING APPLICATIONS THAT USE PACKAGES .....
- USING THE PACKAGE COLLECTION EDITOR .....
- WEAK PACKAGING .....
- ADDITIONAL COMMENTS ABOUT PACKAGES .....
- OVERVIEW OF OBJECTS** .....
- FROM RECORD TO CLASS .....
- ENCAPSULATION AND MEMBER VISIBILITY .....
- Private Members* .....
- Protected Members* .....
- Public Members* .....
- Published Members* .....
- DEFINING THE RUNTIME INTERFACE .....
- INHERITANCE AND POLYMORPHISM .....
- Reusing Code Through Inheritance* .....
- Generic Object Reference Through Polymorphism* .....
- INTRODUCTION TO COMPONENT CREATION** .....
- COMPONENT DESIGN GUIDELINES .....
- A SIMPLE COMPONENT EXAMPLE: DEFINING NEW PROPERTY DEFAULTS** .....
- USING THE COMPONENT WIZARD .....
- OVERRIDING A METHOD .....
- IMPLEMENTING THE OVERRIDDEN CONSTRUCTOR .....
- TESTING A NEW COMPONENT .....
- INSTALLING THE COMPONENT .....
- CREATING THE DESIGN-TIME PACKAGE** .....
- CONFIGURING THE PACKAGE .....
- AN EXAMPLE WITH PROPERTIES** .....
- DEFINING MEMBER FIELDS .....
- DEFINING PROPERTIES .....
- DEFINING PROPERTY CATEGORIES .....
- DEFINING METHODS .....
- Static Methods* .....
- Abstract Methods* .....
- Virtual and Dynamic Methods* .....
- OVERRIDING EXISTING METHODS .....
- IMPLEMENTING THE OVERRIDDEN METHODS .....
- CREATING EVENT PROPERTIES .....
- FINISHING THE COMPONENT: ATTENTION TO DETAIL .....
- Synchronizing Property Values* .....
- Writing Better Event Properties* .....
- RAISING PROPERTY VISIBILITY** .....
- RAISING PROPERTY VISIBILITY EXAMPLE .....
- REINTRODUCED METHODS** .....
- METHODS AND THE IMPLIED CONTRACT** .....

- ADVANCED COMPONENT DESIGN ISSUES .....**
- DESIGN-TIME BEHAVIOR .....
- Detecting Design Time* .....
- PUBLISHING SUBCOMPONENTS .....
- COMPONENT MESSAGE HANDLING .....
- Delphi's Message Handling* .....
- How Messages Are Associated with Methods*.....
- A Message Handling Example* .....
- CREATING DATA-AWARE COMPONENTS .....
- Using Datalinks* .....
- Packages with Data-Aware Controls*.....
- DISTRIBUTING PACKAGE COLLECTIONS.....
- PROPERTY EDITORS.....**
- DERIVING A PROPERTY EDITOR.....
- OVERRIDING INHERITED METHODS .....
- REGISTERING A PROPERTY EDITOR .....
- ADDING A PROPERTY EDITOR TO THE COMPONENT LIBRARY .....
- REPLACING A DEFAULT PROPERTY EDITOR .....
- CREATING COMPONENT EDITORS .....**
- DECLARING THE NEW COMPONENT EDITOR.....
- IMPLEMENTING OVERRIDDEN METHODS.....
- REGISTERING A COMPONENT EDITOR .....
- OVERVIEW OF THE PROPERTY EDITOR AND COMPONENT EDITOR WIZARDS.....**
- INSTALLING THE WIZARDS.....
- USING THE COMPONENT EDITOR EXPERT .....
- USING THE PROPERTY EDITOR WIZARD.....
- APPENDIX A: UNDERSTANDING OBJECTS .....**
- A CLASS DEFINED .....
- Class Contents* .....
- Definition Visibility*.....
- UNITS IN DEPTH .....
- The Interface Section*.....
- The Implementation Section* .....
- The Initialization Section*.....
- The Finalization Section*.....
- Types of Units* .....

Borland, the Borland Logo, and Delphi are trademarks or registered trademarks of Borland Software Corporation in the United States and other countries. Other product and company names mentioned herein may be the trademarks of their respective owners

For more information about training, consulting, mentoring or development, please contact Jensen Data Systems, Inc. by phone at 281-359-3311 or by email at info@JensenDataSystems.com.