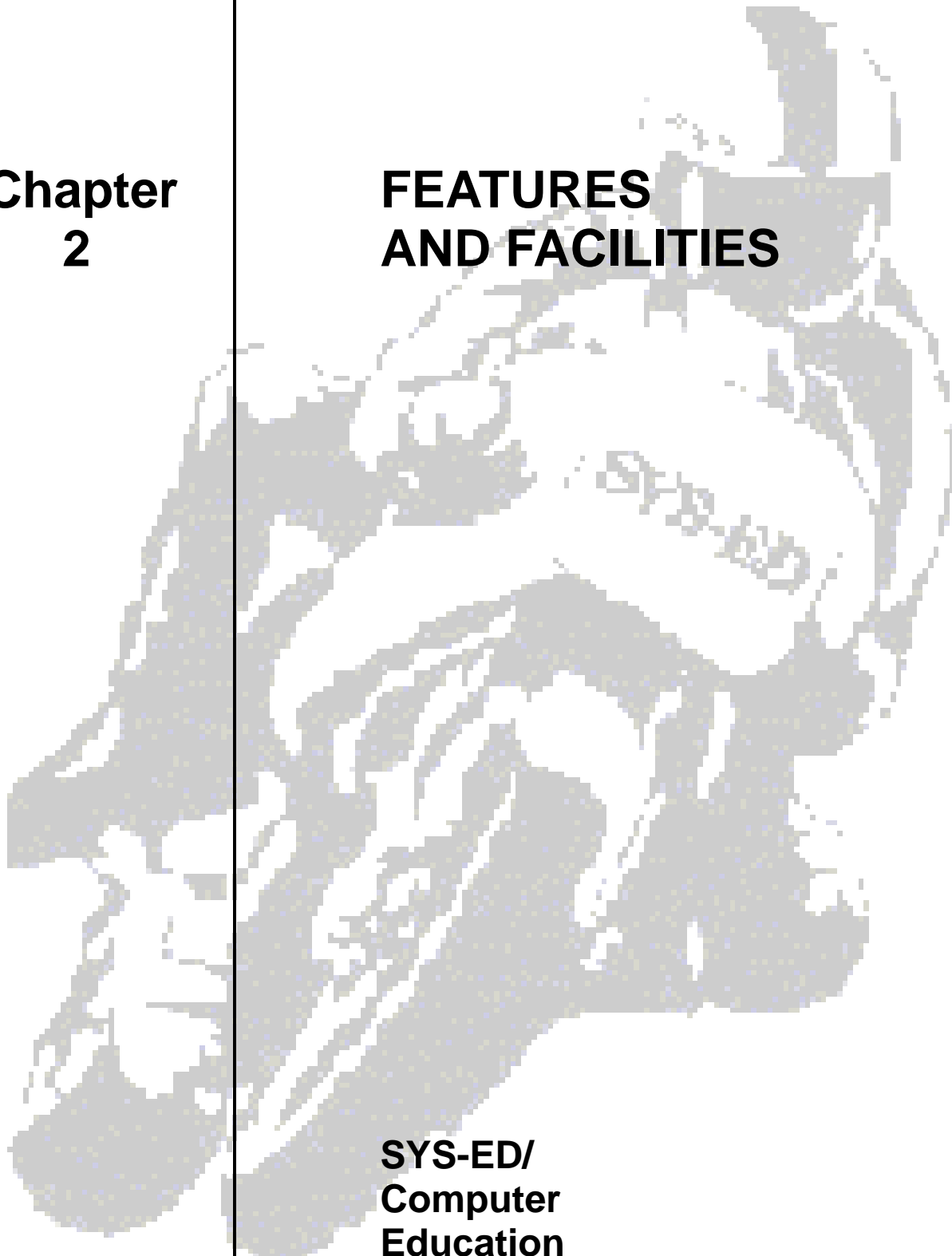


**Chapter
2**

**FEATURES
AND FACILITIES**

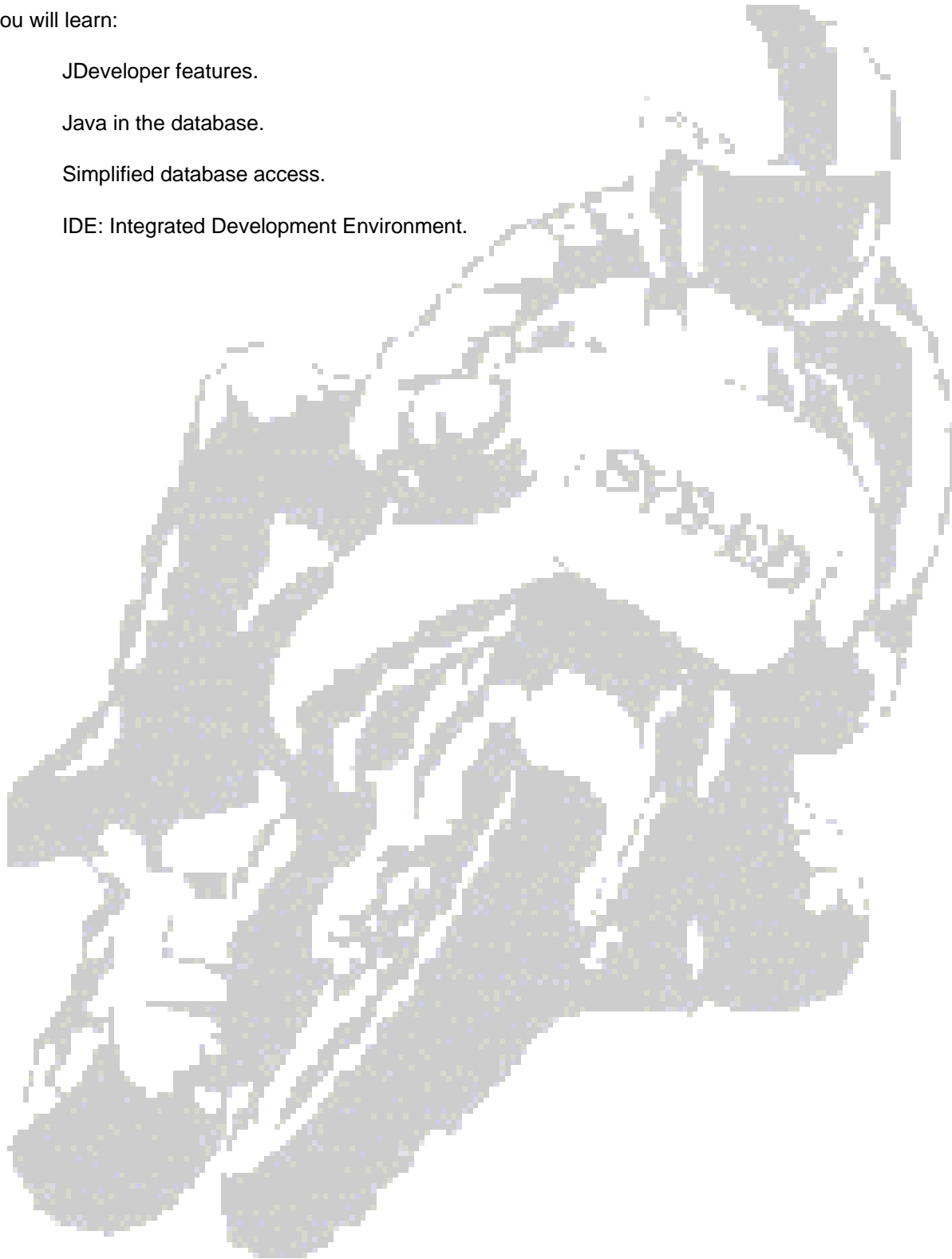


**SYS-ED/
Computer
Education
Techniques, Inc.**

Objectives

You will learn:

- JDeveloper features.
- Java in the database.
- Simplified database access.
- IDE: Integrated Development Environment.



1 JDeveloper Features

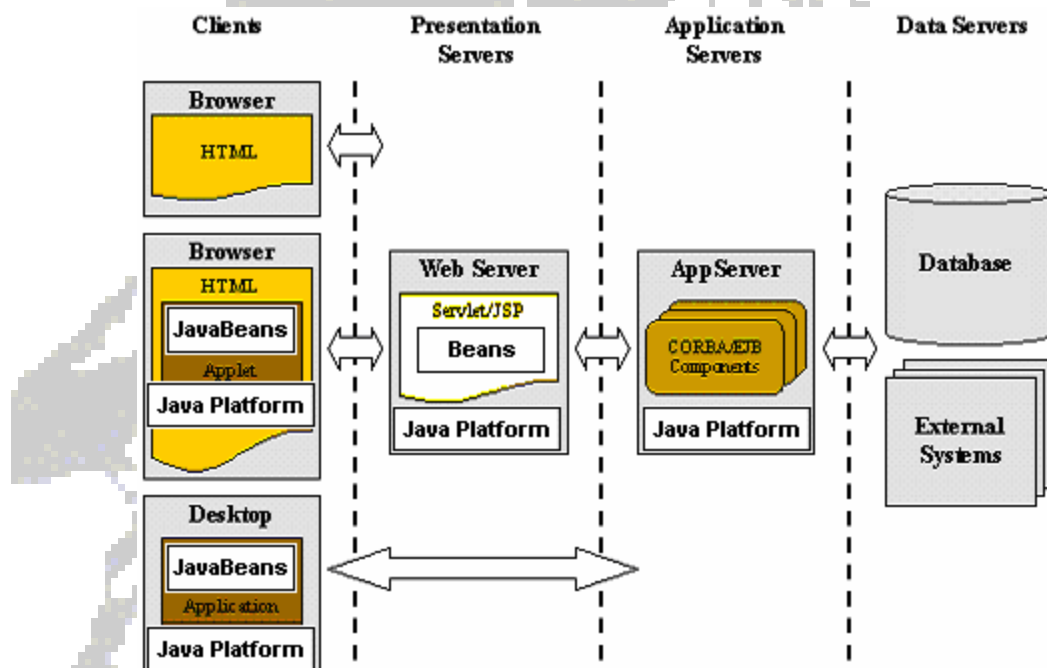
JDeveloper is an integrated development environment for creating multi-tier Java applications.

JDeveloper is used to develop, debug, and deploy:

- Java client applications.
- Dynamic HTML applications.
- Web and application server components.
- Database stored procedures based on industry-standard models.

1.1 Logical Architecture of Applications Built with JDeveloper

This architecture can be physically deployed to a tier configuration of the programmer's choice.



1.2 Oracle: BC4J - Business Components for Java

Oracle BC4J: Business Components for Java supports the standard EJB and CORBA deployment architectures.

BC4J simplifies the development, delivery, and customization of Java business applications for the enterprise. Oracle Business Components for Java is an application component framework which provides developers with a set of reusable software building blocks that manage common facilities required to:

- Author and test business logic in components which integrate with relational databases.
- Reuse business logic through multiple SQL-based views of data.
- Access and update the views from servlets, JavaServer Pages (JSPs), and thin-Java Swing clients.
- Customize application functionality in layers without requiring modification of the delivered application.

By eliminating the coding and testing of common application components, Oracle Business Components for Java allows application developers to concentrate on implementing business solutions.

Behavior provided by the Business Components framework can be overridden in the domain-specific components with a few strategic lines of code.

1.3 Web Application Development

JDeveloper can be used to build and deploy server-side Java applications that deliver dynamically generated HTML to any client running a browser.

Java Servlets	Servlets are portable Java classes that service HTTP requests and dynamically generate HTML. Servlets are portable across platforms and tiers and can be scaled as the needs of your enterprise expand. Servlets are faster than CGI scripts and include integrated support for running in a multithreaded environment. JDeveloper provides wizards to create servlets and an integrated servlet engine for testing and debugging.
JavaServer Pages	JSP: JavaServer Pages technology separates the user interface from dynamic content generation, enabling designers to change the overall page layout without altering the dynamic content. JSP technology supports a reusable component-based design.

JDeveloper provides tools for developing, debugging, and deploying JavaServer Pages. There are wizards to create JavaServer Pages.

- The JDeveloper JSP Element Wizard provides an easy way to add JDeveloper's data tags or predefined web beans to the code.
- JSP 1.1 compliant custom data tags provide access to Business Components for viewing, editing, and full DML control.
- Other tags provide InterMedia support for displaying visual and audio formats.

JDeveloper also provides a wizard to create new web beans. These web beans can be reused in JSPs and provide a convenient way to package common HTML generation logic.

1.4 Data-Aware Controls

JDeveloper includes data-aware controls that conform to the InfoBus industry standard. These controls provide for form-based access to the Oracle database without the need to write any JDBC (Java Database Connectivity) code.

Data-aware controls can be added to a form by dragging and dropping from the JDeveloper tool palette. The transactional behavior and connection information is set in a graphical property editor.

The controls are implemented as JavaBeans and are extensions to the JFC; this means that developers can extend the supplied elements. Applications can be deployed using these controls in either two- or three-tier environments without making modifications.

The Data Form Wizard can be used to create single-table or master-detail forms. After stepping through the Connection Editor, developers can select database tables or Business Components, as well as specify join conditions. The generated forms, which use the InfoBus data-aware controls, require no further coding.

1.5 Component-Based Development

JDeveloper includes GUI-based tools for making intricate code element changes to any class by filling out appropriate fields. With the GUI-based tools, it is easy to add, edit, or delete important elements of any class, including fields, methods, properties, BeanInfo, and event sets.

JDeveloper provides wizards for getting the JavaBean, BeanInfo, Property Editors, Customizers, and Custom Events started.

- The Class Designer is used for making refinements.
- There is a Deployment Wizard for deploying applications.
- The JDeveloper Component Palette can be used to add JavaBeans to any page on the Palette.
- When a JavaBean is created, JDeveloper generates only the requested class and does not generate hidden interfaces or base classes.

1.6 Java Language Support

JDeveloper generates pure Java there by providing for portability and interoperability by adhering to all Java language specifications including JDBC, SQLJ, Servlets, JSP, EJB, JTS, RMI, JARs, and serialization.

JDeveloper ships with the Java 2 SDK version 1.2 and supports JFC/Swing. Since multiple JDKs are available, JDeveloper also provides a feature for switching between available JDKs.

2 Java in the Database

JDeveloper can be used to create Java stored procedures, Enterprise JavaBeans, and CORBA server objects. The programmer can focus on the business logic, while it takes care of the CORBA and Enterprise JavaBean requirements.

2.1 Java Stored Procedures

Java stored procedures can be created and deployed with JDeveloper. Java stored procedures run in the DBMS memory space; this means that complex database functions, as well as repetitive tasks, will execute faster than if they when they are run on the client side. Java stored procedures can be accessed using SQL statements with the standard SQL tools without making modifications.

JDeveloper can deploy any Java class as a collection of stored procedures. When the methods are selected to expose in the database, JDeveloper:

- Creates the appropriate SQL wrappers.
- Determines dependencies.
- Creates the JAR package.
- Performs the installation in Oracle.

JDeveloper also provides an integrated browser for browsing an Oracle database for Oracle Objects, Java procedures, and other procedures.

2.2 Enterprise JavaBeans

Industry-standard Enterprise JavaBeans (EJBs) can be created and deployed to EJB servers and Oracle Application Server. The JDeveloper EJB Wizard creates new enterprise Beans or turns existing Java classes into enterprise Beans. The EJB wizard generates the required EJB deployment descriptor, EJB Home interface, and Remote interface.

At deployment time, JDeveloper works together with Oracle to make EJB deployment straightforward. JDeveloper also offers assistance with the creation of Java client classes that will call the deployed EJB. When an EJB is deployed to Oracle8i, the server generates a remote class that acts as a proxy for the EJB in the client code.

2.3 CORBA Objects

JDeveloper provides the facilities for creating and deploying any Java class as a server-side CORBA object without requiring knowledge of low-level CORBA programming. The JDeveloper IDE includes deployment facilities that can generate IDL and determine packaging dependencies. These files are then bundled into a JAR file and deployed to a CORBA server such as Oracle or Oracle Application Server.

3 Database Access - Simplified

The Oracle Business Components for Java framework will provide the required database access functionality. JDeveloper also provides tools for when low-level access to databases is needed.

3.1 Integrated JDBC

For interfacing with the Oracle database, JDeveloper provides several JDBC drivers:

Oracle Thin JDBC	A pure Java driver that is best suited to applets. It is a Type 4 driver that uses Java sockets to connect directly to Oracle. Written entirely in Java, this driver is platform-independent.
Oracle-OCI	A Type 2 JDBC driver that provides the best access for Java applications running in the middle tier. This driver provides an implementation of the JDBC interfaces that uses the OCI (Oracle Call Interface) to interact with an Oracle database.
Sun ODBC-JDBC Bridge	This driver allows JDBC access from any ODBC compliant database and is best for implementations using a non-Oracle database

JDeveloper also allows developers to use of any third-party JDBC drivers.

3.2 Integrated SQLJ Translator and Debugger

JDeveloper can also use SQLJ programs for low-level database access. A SQLJ program is a Java program containing embedded static SQL statements that comply with the SQLJ language reference syntax. The JDeveloper compiler includes an integrated SQLJ translator and debugger which allows SQL statements to be embedded directly in Java code and debugged as inline source.

SQLJ statements are translated to standard JDBC statements before normal Java compilation; therefore a proprietary language extension is not introduced. The translator may optionally check the syntax of the SQL code, as well as verify it against the actual database schema before including it in JDBC calls. Error messages subsequently direct the developer to the statements in question. The SQL code is debugged inline in the same manner as standard Java code.

SQLJ, is an industry standard which will ensure higher code quality by verifying SQL calls at compile time and also improves developer productivity by reducing the number of lines of database code.

3.3 Connection Editor and Database Browser

All of JDeveloper's wizards and deployment tools include a graphical database connection editor that facilitates JDBC driver selection, username/password entries, and browsing of the actual database schema. The generated configuration is converted to JDBC code, and dependent JDBC libraries are automatically imported to the project. JDeveloper remembers database connections for convenient reuse.

JDeveloper also provides an integrated UI for browsing Oracle schemas to assist database programming.

3.4 JPublisher Wizard

The JPublisher Wizard makes it easier to use Oracle objects in Java programs by automatically generating the appropriate Java class definitions.

Classes generated by JPublisher include methods that convert data from SQL to Java and from Java to SQL. These classes also include methods that get and set attributes of the objects. JPublisher generates Java classes that map to Oracle objects, while also providing the flexibility to extend the generated classes to suit custom needs.

The JDeveloper database browser provides for the navigation of the contents of database schemas, surfacing stored procedures, and Oracle objects. The JPublisher Wizard can be started from the database browser to generate Java wrapper classes for Oracle objects and stored procedures.

4 IDE: Integrated Development Environment

JDeveloper is a visual programming tool with an integrated project navigator, code editor, debugger, and compiler. It also provides wizards, form designers, and property and event editors to automate many programming tasks.

Feature	Explanation
Two-way Technology	Keeps code and design synchronized as a developer works. No compilation is required: a change made in the RAD-style designer is reflected in the code. Similarly, changes to the code are immediately displayed in the Visual Designer.
Unicode-Enabled Compiler	Automatically verifies dependencies in compilation, and error messages link to the section of code in question. This is true in both standard Java and SQLJ.
Docking MDI Interface	This is a Multiple Document Interface which allows developers to switch among editors and designers. The Navigator, Property Inspector, and debugging message windows are dockable palette windows.
Visual Debugger	Supports breakpoints, evaluations, watches, and a wide variety of control flow commands. The visual debuggers support both JDK 1.1 and Java 2.
Multiple Project Management and Libraries	Allows developers to manage many projects, each with their own compilation, run, and debug properties, within one workspace.
CodeInsight	When the editor window is open for coding, the JDeveloper IDE optionally displays coding tips, including parameter lists and class members. A package explorer lists all available packages on the current classpath.
JDeveloper Addin API	A collection of Java APIs for extending the Oracle JDeveloper IDE. With the Addin API, developers gain access to the internals of JDeveloper, providing for the extension of IDE with additional user-created tools or to provide integration with third party tools.
CodeCoach	JDeveloper includes a set of tools that analyze Java code and make suggestions on how to improve its quality and performance. CodeCoach analyzes the code while it is running and generates a detailed list of suggestions.