# ar It oal training

Arkoa acknowledges all registered trademarks.

Arkoa reserves the right to make changes to these materials without prior notice.

© 2002 Arkoa, Inc.

#### Who is Arkoa?

Headquartered in Westborough, MA, Arkoa, Inc. is a technical training company offering corporations and individuals integrated learning solutions in the field of information technology.

Our mission is to create and deliver learning solutions that generate economic value and a competitive advantage for our corporate customers.

Clients from market sectors as diverse as telecommunications, finance, education, government, retail and manufacturing share one certainty—they can depend on Arkoa for the in depth technical and essential desktop application training expertise needed to prepare and deliver high quality training to their customers, partners and internal workforces worldwide.

#### **Contact Information**

*World Wide Web* http://www.arkoa.com

## Headquarters

**4 Technology Drive Westborough,** MA 01**581** Tel. (508) 651-9080 Fax (508) 651-9081

Satellite Offices California Texas New York

# Table of Contents by Course Category

#### **Technology Overviews**

Multi-Tier Client/Server Technical Overview	. 1
COM and DCOM Technical Overview	. 2
CORBA Technical Overview	. 3
Internet Technical Overview	. 4
Intranet Technical Overview	. 5
Electronic Commerce: An Introduction	. 6
Object Technology Technical Overview	. 7
Understanding Information Technology	. 8
Windows NT Architecture Technical Overview	. 9
Data Warehouse Technical Overview1	0
Client/Server Networking and Communications Technical Overview1	11

# Professional Skills Workshops

Communication Skills for IT Professionals12Confrontation and Conflict Management Workshop13Customer Care Workshop14Effective Writing for IT Professionals15Facilitation Skills Workshop16Influencing Skills Workshop17Interviewing Skills for IT Professionals18Managing Change Within the IT Organization19Negotiating Skills for IT Professionals20Presentation Skills Workshop21Problem Solving and Decision Making Workshop23Recruitment Interviewing Workshop23Team Building for IT Professionals24Train the Technical Trainer Workshop25		
Confrontation and Conflict Management Workshop13Customer Care Workshop14Effective Writing for IT Professionals15Facilitation Skills Workshop16Influencing Skills Workshop17Interviewing Skills for IT Professionals18Managing Change Within the IT Organization19Negotiating Skills for IT Professionals20Presentation Skills Workshop21Problem Solving and Decision Making Workshop23Team Building for IT Professionals24Train the Technical Trainer Workshop25	Communication Skills for IT Professionals	. 12
Customer Care Workshop14Effective Writing for IT Professionals15Facilitation Skills Workshop16Influencing Skills Workshop17Interviewing Skills for IT Professionals18Managing Change Within the IT Organization19Negotiating Skills for IT Professionals20Presentation Skills Workshop21Problem Solving and Decision Making Workshop22Recruitment Interviewing Workshop23Team Building for IT Professionals24Train the Technical Trainer Workshop25	Confrontation and Conflict Management Workshop	. 13
Effective Writing for IT Professionals15Facilitation Skills Workshop16Influencing Skills Workshop17Interviewing Skills for IT Professionals18Managing Change Within the IT Organization19Negotiating Skills for IT Professionals20Presentation Skills Workshop21Problem Solving and Decision Making Workshop22Recruitment Interviewing Workshop23Team Building for IT Professionals24Train the Technical Trainer Workshop25	Customer Care Workshop	. 14
Facilitation Skills Workshop16Influencing Skills Workshop17Interviewing Skills for IT Professionals18Managing Change Within the IT Organization19Negotiating Skills for IT Professionals20Presentation Skills Workshop21Problem Solving and Decision Making Workshop22Recruitment Interviewing Workshop23Team Building for IT Professionals24Train the Technical Trainer Workshop25	Effective Writing for IT Professionals	. 15
Influencing Skills Workshop17Interviewing Skills for IT Professionals18Managing Change Within the IT Organization19Negotiating Skills for IT Professionals20Presentation Skills Workshop21Problem Solving and Decision Making Workshop22Recruitment Interviewing Workshop23Team Building for IT Professionals24Train the Technical Trainer Workshop25	Facilitation Skills Workshop	. 16
Interviewing Skills for IT Professionals18Managing Change Within the IT Organization19Negotiating Skills for IT Professionals20Presentation Skills Workshop21Problem Solving and Decision Making Workshop22Recruitment Interviewing Workshop23Team Building for IT Professionals24Train the Technical Trainer Workshop25	Influencing Skills Workshop	. 17
Managing Change Within the IT Organization19Negotiating Skills for IT Professionals20Presentation Skills Workshop21Problem Solving and Decision Making Workshop22Recruitment Interviewing Workshop23Team Building for IT Professionals24Train the Technical Trainer Workshop25	Interviewing Skills for IT Professionals	. 18
Negotiating Skills for IT Professionals.20Presentation Skills Workshop21Problem Solving and Decision Making Workshop22Recruitment Interviewing Workshop23Team Building for IT Professionals24Train the Technical Trainer Workshop25	Managing Change Within the IT Organization	. 19
Presentation Skills Workshop21Problem Solving and Decision Making Workshop22Recruitment Interviewing Workshop23Team Building for IT Professionals24Train the Technical Trainer Workshop25	Negotiating Skills for IT Professionals	. 20
Problem Solving and Decision Making Workshop22Recruitment Interviewing Workshop23Team Building for IT Professionals24Train the Technical Trainer Workshop25	Presentation Skills Workshop	. 21
Recruitment Interviewing Workshop23Team Building for IT Professionals24Train the Technical Trainer Workshop25	Problem Solving and Decision Making Workshop	. 22
Team Building for IT Professionals24Train the Technical Trainer Workshop25	Recruitment Interviewing Workshop	. 23
Train the Technical Trainer Workshop	Team Building for IT Professionals	. 24
	Train the Technical Trainer Workshop	25

# Systems Development and Testing

# Analysis Business Requirements Workshop 26 Development 27 Client/Server GUI Design Workshop 27 Introduction to Programming 28 Testing 28 Developing Test Plans and Test Cases 30 Object-Oriented Testing Methods 31

# Project Management

Client/Server P	roject Management	t Workshop	32
	roject management		

# C/C++ Programming

C Programming	33
Advanced C Programming	35
C++ Programming	36
Advanced C++ Programming	38
C++ Standard Template Library (STL) Programming	39
Visual C++ (6.0) and the MFC Library	41

#### Databases

Database Design/Development	
Data Warehouse Technical Overview	43
Introduction to SQL	44
Logical Data Modeling	45
Logical Data Modeling and Relational Database Design Overview	46
Relational Database Design	47
MS SQL Server	
Introduction to MS SQL Server 7.0	48
Microsoft SQL Server 7.0 Programming and Optimization	49
MS SQL Server 7.0 Database Administration	50
Sybase	
Introduction to SQL and Transact-SQL	52
Developing with Sybase Transact SQL Server	53
Sybase Adaptive Server Administration	54
Oracle7	
Using Oracle7 SQL*Plus	55
Using Oracle7 PL/SQL	56
Oracle7 Database Administration	57
Using Oracle Forms 4.5	58
Using Oracle Reports 2.5	59
Oracle Developer 2000	60
Oracle8	
Using Oracle8 SQL*Plus	61
Using Oracle8 PL/SQL	62
Oracle8 Database Administration	63

#### Java

Java2 Programming	64
Advanced Java2 Programming	65
Introduction to Web Development with Java	66
Java Foundation Classes (JFC) and Swing Development	67
Java Servlet and JSP Programming	68
JavaBeans Development	70
Enterprise JavaBeans Development	72
Enterprise JavaBeans Development	72 75

#### Internet - Web

Internet - Web Overviews	
Internet Technical Overview	77
Intranet Technical Overview	78
Electronic Commerce: An Introduction	79
Introduction to the Web and Web Development	80
Internet - Web Development	
Introduction to Apache Web Server	81
Effective Web Design	82
HTML Programming	84
HTML Fundamentals with Dreamweaver 4	85
HTML Fundamentals with Homesite 4.5	86
Advanced HTML and DHTML Programming	87
JavaScript for Non-Programmers	88
JavaScript Programming	90
VBScript for the Web	91
Internet - Web Programming	
ActiveX and COM Development	92
CGI Programming with Perl	93
Perl Programming	94

Visual InterDev 6.0 and Active Server Pages	95
Introduction to XML	96
XML Programming	97
XML Programming with Java	98

# Object Technology

Object Technology Technical Overview	100
Object-Oriented Analysis and Design	101
UML Object-Oriented Analysis and Design	102
Requirements Analysis and Modeling with Use Cases	103
Introduction to Rational Rose	104
Advanced Object-Oriented Design Using Patterns and Frameworks	106

# Unix/Linux

ntroduction to Unix	107
Jnix Tools and Utilities	108
Jnix System Administration	110
Advanced Unix System Administration	111
ntroduction to Linux	112
Linux Tools and Utilities	113
inux System Administration	114
Advanced Unix Tools	115
Advanced Programming in a Unix Environment	116

# Windows Development

Windows	Programn	ning
---------	----------	------

Windows NT Architecture Technical Overview	117
Windows 95 API Programming	118
Windows NT API Programming	119
Win32 and Win64 System Programming	120
Visual Basic Client/Server Visual Basic Development Enterprise Development with Visual Basic and ActiveX Introduction to Visual Studio Introduction to Microsoft Message Queue Server	121 122 123 124

# **Desktop Application Courses**

Please call for information on the following courses.

Crystal Reports 7.0, Advanced Crystal Reports 7.0, Introduction Crystal Reports, 8.0 Introduction Excel 2000, Advanced Excel 2000, Introduction Excel 2000, Intermediate Excel 2000, Charting with Excel 2000, Data Analysis Excel 2000, VBA Excel 95, Advanced Excel 95, Intermediate Excel 95, Introduction Excel 95, VBA Excel 97, Advanced Excel 97, Intermediate Excel 97, Introduction Excel 97, New Features Excel 97, Data Analysis Excel 97, VBA FrontPage 2000, Introduction FrontPage 98, Advanced FrontPage 98, Introduction HTML Authoring, Advanced HTML Authoring, Introduction Internet/Explorer 4.0, Introduction Internet/Explorer 5.0, Introduction Javascript, Introduction Javascript, Advanced Notes 4.6, Application Development I Notes 4.6, Application Development II Notes 4.6, Technical User Notes R5, Advanced Notes R5, Introduction Notes R5, Mail and Calendar NT 4.0, Administering NT 4.0, Core Technologies Supporting NT 4.0, Installing & Configuring NT 4.0, Introduction Office 97, Introduction Outlook 2000, Introduction Outlook 97, Introduction Outlook 98, Introduction Personal Computer, Introduction PowerPoint 2000, Advanced PowerPoint 2000, Introduction Powerpoint 95, Advanced Powerpoint 95, Introduction Powerpoint 97, Advanced Powerpoint 97, Introduction Project 4.0, Advanced Project 4.0, Introduction Project 98, Advanced Project 2000, Introduction Quattro Pro 8.0, Intermediate Quattro Pro 8.0, Introduction to Sql Server 6.0, Introduction Sql Server 6.0, System Administration Visual Basic 6.0, Advanced Visual Basic 6.0, Intermediate Visual Basic 6.0, Introduction Visual Basic 6.0, New Features

Windows 2000, Introduction Windows 95 for Windows Users Windows 2000 for Windows Users Windows 98, Installation & Configuration Windows 98, Introduction Windows 98, Supporting Word 2000, Advanced Word 2000, Intermediate Word 2000, Introduction Word 95, Advanced Word 95, Intermediate Word 95, Introduction Word 95, New Features Word 97, Advanced Word 97, Intermediate Word 97, Introduction Word 97, New Features Wordperfect 8.0, Intermediate Wordperfect 8.0, Introduction

# Table of Contents — Alphabetical

ActiveX and COM Development	92
Advanced C Programming	35
Advanced C++ Programming	38
Advanced HTML and DHTML Programming	87
Advanced Java2 Programming	65
Advanced Object-Oriented Design Using Patterns and Frameworks	06
Advanced Programming in a Unix Environment1	16
Advanced Unix System Administration1	11
Advanced Unix Tools 1	15
Business Requirements Workshop	26
C Programming	33
C++ Programming	36
C++ Standard Template Library (STL) Programming	39
CGI Programming with Perl	93
Client/Server GUI Design Workshop	27
Client/Server Networking and Communications Technical Overview	11
Client/Server Project Management Workshop	32
Client/Server Visual Basic Development1	21
COM and DCOM Technical Overview	2
Communication Skills for IT Professionals	12
Confrontation and Conflict Management Workshop	13
CORBA Technical Overview	3
Customer Care Workshop	14
Data Warehouse Technical Overview10,	43
Developing Test Plans and Test Cases	30
Developing with Sybase Transact SQL Server	53
Effective Web Design	82
Effective Writing for IT Professionals	15
Electronic Commerce: An Introduction	79
Enterprise Development with Visual Basic and ActiveX1	22
Enterprise JavaBeans Development	72
Facilitation Skills Workshop	16
HTML Fundamentals with Dreamweaver 4	85
HTML Fundamentals with Homesite 4.5	86
HTML Programming	84
Influencing Skills Workshop	17
Internet Technical Overview	77
Interviewing Skills for IT Professionals	18
Intranet Technical Overview	78
Introduction to Apache Web Server	81
Introduction to Linux1	12
Introduction to Microsoft Message Queue Server 1	24
Introduction to MS SQL Server 7.0	48
Introduction to Programming	28
Introduction to Rational Rose1	04
Introduction to SQL	44
Introduction to SQL and Transact-SQL	52
Introduction to the Web and Web Development	80
Introduction to Unix1	07
Introduction to Visual Studio 1	23
Introduction to Web Development with Java	66
Introduction to XML	96
Java Foundation Classes (JFC) and Swing Development	67
Java Servlet and JSP Programming	68
Java2 Programming	64
JavaBeans Development	
	/0
JavaScript for Non-Programmers	70 88

Linux System Administration
Linux Tools and Utilities
Logical Data Modeling
Logical Data Modeling and Relational Database Design Overview
Managing Change Within the IT Organization
Microsoft SQL Server 7.0 Programming and Optimization
MS SQL Server 7.0 Database Administration 50
Multi-Tier Client/Server Technical Overview
Negotiating Skills for IT Professionals
Object Technology Technical Overview
Object-Oriented Analysis and Design
Object-Oriented Testing Methods
Oracle Developer 2000
Oracle7 Database Administration
Oracle8 Database Administration
Perl Programming
Presentation Skills Workshop
Problem Solving and Decision Making Workshop
Recruitment Interviewing Workshop
Relational Database Design
Requirements Analysis and Modeling with Use Cases
Sybase Adaptive Server Administration
Team Building for IT Professionals
Train the Technical Trainer Workshop
UML Object-Oriented Analysis and Design
Understanding Information Technology
Unix Fundamentals
Unix System Administration
Unix Tools and Utilities 109
Using Oracle Forms 4.5
Using Oracle Reports 2.5
Using Oracle7 PI /SOI
Using Oracle7 SOI *Plus 55
Using Oracle8 PL/SOI
Using Oracle8 SOI *Plus 61
VBScript for the Web
Visual $C_{++}$ (6.0) and the MEC Library 41
Visual InterDev 6.0 and Active Server Pages
Win32 and Win64 System Programming 120
Windows 95 API Programming
Windows NT API Programming 119
Windows NT Architecture Technical Overview 9 117
XMI Programming
XMI Programming with Java
XML Programming with Java

# Multi-Tier Client/Server Technical Overview

#### 2 days

#### Description

This is a high-level, practical overview of the client/server computing environment. This course focuses on the multi-tier client/server environment including 3-tier and n-tier client/server and Web architectures. This course is normally delivered in two days with hands-on exercises. It can be delivered in one day in a condensed lecture-only format.

#### Audience

IT professionals interested in multi-tier client/server application development including project managers, team or group leaders, analysts, designers, programmers, and support personnel.

#### Prerequisites

Familiarity with client/server computing systems and programming concepts.

#### **Classroom Requirements**

Overhead projector, whiteboard, flipcharts.

#### Topics

#### Introduction to Client/Server Applications

- Multi-tier application basics
- Multi-tier application influences
- Multi-tier application styles
- Multi-tier application challenges

#### Architecting Multi-Tier Client/Server Applications

- Basic multi-tier application architecture
- Multi-tier architectural guidelines and options
- Making architectural choices

#### **Building Multi-Tier Client/Server Applications**

- Middleware basics
- Multi-tier middleware
- Building the multi-tier application

#### Format

#### 🖪 Hands-on

Written Exercises

#### Objectives

After completing this course, participants should be able to:

- Recognize the driving forces for multi-tier client/server technology
- Describe multi-tier client/server concepts
- Describe the common types of client/server and multi-tier client/server applications
- Describe the client/server software development lifecycle
- Identify client/server and multi-tier analysis and design considerations
- Identify client/server and multi-tier coding and testing considerations
- Describe how multi-tier client/server development differs from traditional client/server development

#### Analysis and Design of Multi-Tier Applications

- Analysis and design principles
- Analysis techniques for multi-tier applications
- Design techniques for multi-tier applications

#### **Developing Multi-Tier Client/Server Applications**

- Multi-tier development principles
- Development considerations
- Development tools
- Approaches to testing

#### **Operational Considerations for Multi-Tier Applications**

- Preparation for deployment
- Operational considerations

Course # 11-0601

# COM and DCOM Technical Overview

#### 1 day

#### Description

This is a high-level, practical overview of COM and Distributed COM architecture and development implications. After building an understanding of COM design, this course explores alternatives for leveraging server executables and components in traditional client/server environments and advanced, multi-tier, distributed environments. It also provides advice on the strengths and weaknesses of current development tools for building the types of servers, components, and clients discussed in the course.

#### Audience

COM, ActiveX, and OLE programmers who need an introduction to the COM technologies. Managers and other IT professionals who need an understanding of COM technologies.

#### Prerequisites

Familiarity with object-oriented languages and technologies and the Windows platform.

#### Classroom Requirements

Overhead projector, flipcharts

#### Topics

#### Introduction

#### Definitions

- What is object technology?
- Distributed components

#### Introduction to COM and DCOM

♦ What is COM?

#### **COM Programming**

- Implementing COM
- What is an active control?
- What is the ATL?
- COM development strategies

#### Format

#### Presentation

#### Objectives

After completing this course, participants should be able to:

- Describe, at the conceptual level, COM and related technologies
- Plan the utilization of COM technologies in IT/IS projects
- Address the implications of COM and DCOM technologies to systems management
- Evaluate and select development tools and environments

#### **COM Interoperability**

- Introduction to CORBA
- CORBA and DCOM compared
- Interoperability and interworking

#### COM+

- Windows 2000
- Microsoft Transaction Server (MTS)
- ♦ Introduction to COM+

2

# **CORBA** Technical Overview

#### 1 day

#### Description

The CORBA standard has evolved to the state where the promise of distributed object component computing over heterogeneous networks can be realized. This overview describes the technology CORBA uses, how vendors have implemented the standard, and ways for CORBA environments to interact with others.

#### Audience

Application developers and designers, IT managers and other professionals who need to understand CORBA technologies and potential implications.

#### Prerequisites

Attendance of Arkoa's *Object Technology Technical Overview* or a similar level of understanding is required.

#### **Classroom Requirements**

Overhead projector, whiteboard, flipcharts.

#### Topics

#### Introduction

#### The CORBA Standard

- History of CORBA and OMG
- CORBA 1.x revisions
- CORBA 2.0

#### CORBA Components

- Defining terms and components
- ORBs
- IDL and repositories
- ♦ BOA
- Object references and invocations
- ♦ IIOP
- CORBA services
- Developing CORBA applications
- Components versus objects

#### Format

#### Presentation

#### Objectives

After completing this course, participants should be able to:

- Describe the major architectural components and history of the CORBA standard
- Contrast the features of CORBA and COM/DCOM
- Identify current tools and support for managing, designing, and implementing CORBA environments
- Describe how to integrate CORBA environments with other component standards and with Java

#### **CORBA** and Java

- CORBA capabilities
- Enterprise JavaBeans
- Emerging standards

#### Vendor CORBA Products

- Inprise's VisiBroker
- ♦ IBM's SOM
- Iona's Orbix

#### Interoperability

- ORB-to-ORB bridging
- Microsoft's DCOM
- Web/internet implication

# Internet Technical Overview

#### 1 day

#### Description

This technical overview provides a high-level introduction to the Internet. It describes Internet technologies and applications and discusses the benefits of the Internet as well as the issues that need to be addressed. It explains how to make productive use of the Internet in personal and/or business life. If the classroom is fitted with workstations and access to the Internet, this overview includes a demonstration and some hands-on exercises.

#### Audience

echnonlogy

Overviews

Technical managers and IT professionals who have little or no Internet experience.

#### Prerequisites

Familiarity with basic computer terminology is required.

#### **Classroom Requirements**

Overhead projector and flipcharts.

#### Topics

#### Introduction

- What is the Internet?
- The history of the Internet
- What is the World Wide Web?
- The history of the World Wide Web
- Internet connectivity
- Internet culture
- What is an intranet?

#### Internet Technologies

- Network technologies: transmission protocols and Internet access
- World Wide Web technologies: URLs and HTTP
- Web content creation technologies: HTML and DHTML
- Data access
- Security

#### Format

- Presentation
- ▲ Demonstration

#### Objectives

After completing this course, participants should be able to:

- Use and understand Internet terminology
- Explain how the Internet relates to the World Wide Web
- · Describe the benefits and issues involved in connecting to the Internet
- Describe the history of the Internet
- Describe Internet technologies and standards
- Describe popular Internet applications
- Use a popular Web browser

#### **Internet Applications**

- Electronic mail
- ftp and Telnet
- News
- Chat
- The World Wide Web: browsers, Web site management, and applications
- Benefits and Issues
- Business benefits and electronic commerce
- Issues

# Intranet Technical Overview

#### 1 day

#### Description

This high-level overview explains how intranets can benefit work groups and companies. It describes the enabling technologies and popular products and explains the issues that need to be considered when planning, preparing for, or implementing an intranet.

#### Audience

IT managers and professionals interested in developing or assessing an intranet.

#### Prerequisites

Familiarity with basic computer terminology is required. High level understanding of Internet concepts is helpful.

#### **Classroom Requirements**

Overhead projector, flipcharts

#### Topics

#### What is an Intranet?

- Web and Internet technology basics
- The Internet versus intranets
- Intranet content
- Intranet building blocks ٠

#### The Benefits of an Intranet

- Intranet uses and benefits
- ٠ Intranets and groupware
- Intranet readiness factors

#### The Business Case for an Intranet

- Intranet start-up costs
- Cost saving areas
- Preparing an intranet cost justification

#### **Developing Intranet Applications**

- Overview of an intranet application
- Document publishing ٠
- Intranet application development

#### Extranets

- Why are extranets needed?
- Issues raised by extranets
- Typical extranet applications

#### Intranet Components and Architectures

- Basic intranet components
- Web browser functionality ٠
- Web server functionality ٠
- Choosing Web browsers and servers ٠
- ٠ Web architectures

#### Format

#### Presentation

#### Objectives

After completing this course, participants should be able to:

- · Describe intranets, their history, and their relationship to the Internet
- Identify the benefits and issues involved in implementing an intranet ٠
- Describe intranet architectures and architectural components ٠
- Explain intranet security threats, solutions, and issues
- Describe intranet applications and what an intranet can do ٠
- Recognize the differences between groupware and an intranet

#### Planning for an Intranet

- Planning steps
- Strategy
- Analysis and design
- Implementation
- Creating a plan
- Marketing and evaluation

#### **Building an Intranet**

- Evaluate and select components
- Web server software and browsers ٠
- Other services ٠
- Installing components ٠
- Laying out the Web site
- Web page design ٠
- Building the Web site ٠
- Implementing policies
- Setting up templates

#### Intranet Security

- Security requirements, policy, and mechanisms
- Data privacy mechanisms

#### Managing the Intranet

- Management and administration
- Webmaster and Web administrator roles
- Advanced Intranet Applications

#### Transaction processing

- Integration with groupware and existing systems
- Application design

Course # 11-0100

#### 1 day

#### Description

The term "electronic commerce" means different things to different people. This course reviews these definitions and provides a summary of how business can be conducted electronically. This course also explores the business and technology considerations that are making the World Wide Web a popular vehicle for conducting business.

#### Audience

echnonlogy

**Overviews** 

Anyone interested in electronic commerce, including technical managers and IT professionals.

#### Prerequisites

Familiarity with basic computer terminology is required. Familiarity with Internet/World Wide Web concepts is desirable.

#### Format

- Presentation
- 🖴 Hands-on

#### Objectives

After completing this course, participants should be able to:

- Identify the principal components of electronic commerce
- Compare traditional and electronic ways of doing business
- Recognize commonly used protocols and technologies utilized in an ecommerce site
- Understand differences between business-to-business and business-toconsumer e-commerce
- Differentiate between electronic payment protocols
- Identify large and small business e-commerce solutions
- Understand security risks and alternatives for securing e-commerce systems
- Plan the creation and management of an electronic business

#### Topics

#### **Evolution of Electronic Commerce**

- Traditional versus electronic business
- History and growth of electronic commerce
- E-commerce models

#### Infrastructure for Electronic Commerce

- Networking technologies and protocols
- Internet architectures
- Internet service providers and services
- Enabling tools
- Selling and billing infrastructures
- Purchasing and order management infrastructures

#### **Electronic Commerce Applications**

- Business to business
- Business to consumer
- Industries and applications
- Case studies
- Legal issues

#### **Electronic Payment Systems**

- Types of business transactions
- Forms of payment
- E-commerce standards
- ◆ EDI, EFT, and CALS
- ♦ SET
- Taxation

#### **Commercial Merchant Solutions**

- Payment infrastructures (CyberCash, CyberSource, CyberTrust)
- Enterprise/large business solutions (IBM, Netscape, Microsoft, Sun)
- Small business solutions
   Security Considerations

- Risk areas
- Consumer privacy
- Securing payment processes
- Encryption
- Authentication
- Virtual Private Networks

#### **Electronic Commerce Business Opportunities**

- Creating awareness
- Marketing/selling products and services

#### Establishing an Electronic Business

- Scalable system design
- Integration with legacy systems
- Audit trails and reporting
- Managing the business

# **Object Technology Technical Overview**

#### 1 day

#### Description

This technical overview provides an introduction to the concepts and benefits of object-oriented techniques. The overview describes the concepts of encapsulation, abstract data types, message passing, inheritance and polymorphism. The differences between object-oriented design and traditional methods are discussed and the important organizational implications of adopting an object-oriented approach are considered.

#### Audience

IT (information technology) managers and staff who are transitioning to an object-oriented software development approach.

#### Prerequisites

An understanding of business applications and familiarity with the process of analysis and design for a typical software development project are helpful.

#### **Classroom Requirements**

Overhead projector, flipcharts

#### Topics

#### Introduction to Object Technology

- Why object technology? Driving forces for object technology Computing context
- Object-oriented concepts
   Objects everywhere
   Object concepts and characteristics
   Classes and their characteristics

#### **Object-Oriented Analysis**

- Analysis methods and notations
   Object-oriented development life cycle
   Incremental model
   Analysis and design methods
- Performing object-oriented analysis Review of analysis techniques Review of analysis notation
- Perform analysis

#### Format

#### Presentation

#### Objectives

After completing this course, participants should be able to:

- Explain the driving forces for object technology and describe its evolution
- Define terms related to object technology and explain fundamental object-oriented concepts
- Define object-oriented analysis and design
- List and describe leading analysis and design methodologies
- Explain object-oriented analysis techniques and the process of objectoriented design
- List and describe tools and languages used for object-oriented development
- Define and describe components and component frameworks
- Describe the roles and skills needed for object-oriented development

#### **OO Design and Development**

- Object-oriented design
   GUI design considerations
   Perform design
- Object-oriented tools
   Languages
   Class libraries

Graphical user interface tools Object-oriented databases Development considerations

#### Implementing Object Technology

- Distributed component frameworks
   Distributed objects
   Object management architecture and the common object
   Compound document framework
   COM/DCOM and OLE
- Building a career in object technology Changing skill requirements in the development life cycle The project team's roles and responsibilities Skills and training

#### Appendix: Object Technology Definitions

Course # 11-0150

# **Understanding Information Technology**

#### 1 day

#### Description

This workshop introduces information technology (IT) to business managers and users. It explains how IT supports business processes and provides a basic understanding of hardware, software, applications, and terminology. It also explains the generic processes within the information systems organizations and how to work with IT. It can be customized for company-specific environments.

#### Audience

lechnonlogy Overviews

Business managers and users who need to know how to leverage IT.

#### Prerequisites

#### None. Classroom Requirements

Overhead projector, flipcharts

#### Topics

#### Introduction

#### **Business Context**

- Business goals and processes
- Types of applications and their uses
- Relationship between IT and business
- Current business/IT problems and their causes

#### Types of Computing

- Mainframes
- Personal computers
- Distributed computing

#### Hardware

- The computer
- Storage devices
- Peripherals
- Purchase criteria
- Hardware requirements

#### Software

- Applications
- Programming
- System software
- The user interface
- Usability

#### Course # 11-0120

#### Format

- Presentation
- Written Exercises
- ♥ Interactive Activities

#### Objectives

After completing this course, participants should be able to:

- Recognize how IT supports business processes
- Describe types of computing
- Describe the basic IT components
- Recognize the capabilities and limitations of IT
- Use IT terminology
- Explain the key success factors for working successfully with IT professionals

#### **Databases and Data**

- File systems and databases
- Data for decision-making
- Data in transaction processing

#### Networks and Client/Server

- Networks and communications
- Network infrastructure
- ♦ Client/server

#### Internet and WWW

- Internet
- Intranet

#### Software Development Life Cycle

- Stages in software development
- Requirements and implementation
- · Methods for development

#### Conclusion

· Key success factors for working with IT

# Windows NT Architecture Technical Overview

#### 1 day

#### Description

This is a high-level, practical overview of the Windows NT architecture. This one-day overview describes each of the important Windows NT subsystems and explains how they interoperate. The course also provides an overview of the Windows NT filesystem, BackOffice Server, and Microsoft's Distributed intraNet Architecture (DNA). The major features of Windows 2000 are explained.

#### Audience

IT professionals, including project managers, team or group leaders, analysts, designers, programmers, and support personnel.

#### Prerequisites

Familiarity with computing systems and programming concepts.

#### **Classroom Requirements**

Overhead projector and flipcharts.

#### Topics

#### Introduction to Windows NT Architecture

- Design goals
- Comparison of editions
- System architecture components
- Tools to view NT internals

#### Windows NT Executive Subsystems

- An object management overview
- A security system overview
- Process management overview
- An LPC facility overview
- A memory management overview
- Win32 GUI subsystem overview
- An I/O system overview
- Cache manager overview

#### Windows NTFS and File Server

- NTFS major features
- Comparison with FAT
- NTFS structure
- The Windows NT filesystem tree
- Windows NT file server

#### Windows DNA

- ◆ A Windows DNA overview
- Windows DNA technologies

#### Format

#### Presentation

#### Objectives

After completing this course, participants should be able to:

- Describe the functionality and components of Windows NT system architecture
- Describe the structure and functionality of Windows NT Executive components
- Describe the structure and operation of Windows NTFS
- Describe the functionality and integration of BackOffice Server products with Windows NT
- Describe the structure and functionality of Windows DNA
- Describe the major new functionality and components of Windows 2000

#### BackOffice Server 4.0

- ◆ A BackOffice products overview
- Windows NT option pack
- Other BackOffice products
- Deployment: Internet Bookstore
- Deployment: Satellite Offices

#### Windows 2000

- Windows 2000 products
- Selected major features

Technonlogy Overviews

#### 1 day

#### Description

Data warehouses are able to turn raw data into business intelligence that supports effective decision making and provides a competitive advantage. However, designing and building a data warehouse requires a new way of thinking and substantial investment. This overview provides a high-level introduction to data warehouse alternatives and issues.

#### Audience

**Technonlogy** 

**Overviews** 

Managers and professionals evaluating or implementing data warehouse technology.

#### Prerequisites

A basic understanding of relational databases is preferred.

#### **Classroom Requirements**

Overhead projector, whiteboard, flipcharts.

#### Topics

#### Introduction

#### **Driving Forces**

- Business pressures
- Technology drivers

#### **Data Warehouse Definitions**

- Why consider a data warehouse?
- What is a data warehouse?
- Benefits and risks

#### **Data Warehouse Capabilities and Applications**

- Data storage
- Operational data
- Informational systems
- Analytic systems
- Tools and their use

#### Format

- Presentation
- Written Exercises

#### Objectives

After completing this course, participants should be able to:

- Describe the business benefits of a data warehouse
- Describe different types of data warehouses
- Identify the front- and back-end components of a data warehouse
- Identify the critical success factors in designing and building a data warehouse

#### **Data Warehouse Architecture Components**

- Data warehouse elements
- Middleware
- Database engines
- Applications
- Datamarts

#### **Data Warehouse Design and Development**

- Defining requirements
- Modeling data
- Cleansing data
- Performing extraction, transformation, and loading
- Maintaining the data warehouse

#### Conclusion

Critical success factors

# Client/Server Networking and Communications Technical Overview

#### 1 day

#### Description

This is a high-level overview that provides a framework for understanding communications and networking. It establishes a common vocabulary and context useful for considering the networking issues of implementing systems in a client/server environment. Business and technical considerations are explored. Key technologies, protocols, network management and security issues are also described.

#### Audience

Anyone concerned with communications and networking, including business managers, senior technical managers, MIS managers, DP managers, senior system designers and analysts, and software engineers.

#### Prerequisites

Familiarity with basic computer terminology is required.

#### **Classroom Requirements**

Overhead projector and flipcharts.

#### Topics

#### Introduction

- Introductions
- Housekeeping and Administration
- Course Agenda
- The OSI Reference Model
- OSI Definition and Goal
- OSI Reference Model Layer descriptions
- OSI Reference Model Protocols
- TCP/IP and the OSI Reference Model
- Microsoft Networking and the OSI Reference Model

#### **Topology Protocols**

- Physical WAN Connections
- LAN Wiring and Topology
- LAN Protocols
- WAN Topology

#### Internetworking

- Defining Internetworks
- Bridging
- Routing
- Switching
- Protocols

#### Format

#### Presentation

#### Objectives

After completing this course, participants should be able to:

- · Describe the context that forms the basis for network computing
- Use the associated vocabulary and definitions
- Identify the characteristics of key technology components
- Describe the services and products that form the basis for building networks

# Technonlogy

#### Network Operating Systems (NOS)

- NOS Functions
- NOS Products
- ♦ Focus on NetWare
- Focus on NT Server
- Unix as a NOS

#### Internet

- ♦ A Brief History
- Internet Services and Protocols
- Internet Connectivity
- Firewalls

#### Security and Network Management

- Security
- Network Management

# **Communication Skills for IT Professionals**

#### 2 days

#### Description

Effective communication skills are the key to influencing others and to establishing and maintaining good relationships. This interactive workshop describes how to achieve mutual understanding and greater clarity in interpersonal communication, identifies and develops communication skills, and provides opportunities for refining skills through role-playing activities.

#### Audience

IT (information technology) and other professionals at all levels.

Maximum number of participants: 12

#### Prerequisites

No prerequisites.

#### **Classroom Requirements**

Room set up as horseshoe and large enough for team activities, or with break-out rooms for team activities.

#### Topics

#### **Introduction**

#### Communication Overview

- Identifying Barriers to Communication
- Defining Communication
- Results of Miscommunication

#### Interpersonal Communication

- Body Language
- Using Body Language to Communicate
- The Pitfalls
- Active Listening
- Verbal Communication
- Choosing and Matching Language
- Appropriate Language

#### Format

- Presentation
- Written Exercises
- ♥ Interactive Activities

#### Objectives

After completing this course, participants should be able to:

- Identify barriers to communication
- Identify verbal and non-verbal communication techniques
- Employ active listening techniques
- Recognize effective telephone techniques
- Employ influencing techniques
- Use appropriate terminology for different audiences

#### **Telephone Skills**

- Effective Telephone Techniques
- Voice Matching
- Telephone Tips
- ♦ Voice Mail Guidelines

#### **Influencing Skills**

- Influencing Factors
- Preparing to Influence
- Influencing Others
- Influence or Manipulation
- Workshop Review
- Review
- Post-Workshop Assessment

# **Confrontation and Conflict Management Workshop**

#### 2 days

#### Description

In today's challenging work environment, individuals are encouraged to take responsibility and make individual contributions. Projects are often undertaken by cross-functional and multi-functional teams which afford many opportunities for creativity. However, there are also many occasions when disagreement and conflict cause frustration and reduce productivity. This workshop identifies situations in which conflict may occur and suggests techniques for handling confrontation and conflict. It also provides exercises for establishing and reinforcing those techniques.

#### Audience

Persons in jobs or roles in which managing conflict and disagreement is important.

Maximum number of participants: 12

#### Prerequisites

Participation in Arkoa's *Communication Skills* and *Negotiating Skills* workshops is helpful.

#### **Classroom Requirements**

Room set up as horseshoe and large enough for team activities, or with break-out rooms for team activities. Overhead projector, flip charts.

#### Topics

#### Introduction

#### Drivers for Confrontation in the Workplace

- Disagreement and conflict
- Changes in the workplace

#### Motivation for Managing Conflict

- Organizational styles and culture
- Why and when people disagree at work

#### Levels of Disagreement

- Leveraging disagreement
- Constructive and creative discussion
- Techniques for analyzing causes and sources of disagreement
- Achieving consensus

#### **Recognizing Personal Style**

- Personal styles of handling disagreement
- Personal styles of handling conflict

#### Format Presentation

- Written Exercises
- Interactive Activities

#### Objectives

After completing this course, participants should be able to:

- Explain the increase in workplace conflict and what forces are driving this increase
- Explain the motivation for managing conflict in the workplace
- Explain and compare disagreement and conflict
- Analyze personal styles and their implications for conflict management
- Apply effective techniques to managing conflict.
- Apply effective communication and negotiating techniques to conflict management
- Explain how to manage difficult individuals within the context of conflict

#### Effective Communication and Negotiation

- Verbal and non-verbal communication
- Win-win negotiation techniques

#### Conflict

- Techniques for managing conflict
- Critical and non-critical issues
- Stakeholders
- Sources of disagreement and dissension

#### Handling Confrontations

- Difficult situations and personalities
- Techniques for managing difficult people

#### Conclusion

Critical success factors

#### 2 days

#### Description

In a highly competitive marketplace, customer care is the key to a company's success. This workshop describes effective customer service techniques and provides opportunities for participants to apply and refine them through role plays and exercises.

#### Audience

Customer service representatives, help desk personnel, and technical support staff

Maximum number of participants: 12

#### Prerequisites

This workshop is a component of Arkoa's Professional Skills Workshops. Follow-on workshops include *Communication Skills*, *Negotiating Skills*, *Interviewing Skills*, *Confrontation and Conflict Management*, and *Problem Solving and Decision Making*.

#### **Classroom Requirements**

Room set up as horseshoe and large enough for team activities, or with break-out rooms for team activities.

#### Format

- Presentation
- Written Exercises
- ♥ Interactive Activities

#### Objectives

After completing this course, participants should be able to:

- Evaluate the importance of effective customer service
- Identify customer expectations and how to manage them
- Recognize typical customer profiles and behaviors
- Describe the importance and impact of taking ownership of customer problems
- Define the professional behaviors needed to manage customer relationships and ensure customer satisfaction
- · Use effective communication skills to obtain and impart information
- Apply techniques for handling difficult customer situations
- Recognize organization and time management needs in effective customer care

#### Topics

#### Introduction

#### **Customer Care Context**

- Defining customer care
- Business implications of customer care
- The importance of customer care
- Customer reactions
- From customer service to customer care
- Types of customer care

#### The Customer

- Customer satisfaction
- Types of customers
- Teamworking and networking
- Customer needs and expectations
- Customer satisfaction factors
- Customer interactions and behaviors
- Interacting with customers
- Responding to customers

#### The Customer Care Professional

- The role of the customer care professional
- Obtaining and clarifying information
- The role of listening
- Telephone skills
- Managing and recording interactions
- Effective voice mail
- Elements of effective communication
- Information management

#### Managing Relationships

- Managing difficult situations
- · Negotiating skills: preparation, negotiation, and follow-through
- Handling customer situations
- Handling conflict

#### Self-Management

- Organizing and managing yourself
- Handling stress
- **Critical Success Factors**
- Personal action plan
- Critical success factors

# **Effective Writing for IT Professionals**

#### 2 days

#### Description

Companies, teams, and individuals are measured by how well they communicate. This interactive workshop helps IT professionals produce more effective documents. It reviews technical writing techniques and processes and provides examples and templates for continuing use after the workshop. Company specifics, such as style guidelines, can be incorporated. Each topic is reinforced by written exercises.

#### Audience

Information technology (IT) professionals who want to improve the effectiveness of their writing. This course can be customized for business professionals or mixed audiences.

Maximum number of participants: 12

#### Prerequisites

Good verbal and written English skills are required.

#### **Classroom Requirements**

Room set up as horseshoe and large enough for individual and group writing activities and reviews. Overhead projector, flipcharts.

#### Topics

#### Introduction

#### **Technical Writing Overview**

- Pre-sales documents and proposals
- System development life cycle documents
- Business documents

#### **Effective Technical Writing**

- Writing basics review: sentence structure, active and passive voice, person, punctuation, using the correct words
- Common mistakes in technical writing

#### Writing for a Target Audience

- Assessing the target audience
- Assessing the audience's needs

#### Format

- Presentation
- Written Exercises

#### Objectives

After completing this course, participants should be able to:

- Describe the types of documents that IT professionals develop and use
- Apply the basic rules of good technical writing and avoid jargon
- Identify the needs of different audiences
- Use outlines for organizing document content
- Summarize key points of verbal or written information
- Use effective presentation formats
- Write effective electronic mail

#### **Using Outlines and Summaries**

- Outline formats and elements
- Elements of effective summaries

#### Writing Effective E-mail

• E-mail message guidelines

#### Using Effective Presentation Formats

- Graphics and other presentation elements
- Presentation guidelines

#### Conclusion

Success factors

# **Facilitation Skills Workshop**

#### 2 days

#### Description

Teams and groups are much more effective if they are led by a skilled facilitator who encourages participation and creativity while maintaining focus and staying within time constraints. A competent facilitator makes review sessions, meetings, and workshops more productive. A facilitator understands how to plan and prepare for group sessions and how to exercise the interpersonal skills that achieve best results. This interactive workshop describes these skills and affords students an opportunity to apply and refine them through extended role playing activities.

#### Audience

Anyone involved in formal or informal facilitation.

Maximum number of participants: 12

#### Prerequisites

An understanding of communication skills is needed—attendance of Arkoa's *Communication Skills for IT Professionals* is helpful.

#### **Classroom Requirements**

Room set up as horseshoe and large enough for team activities, or with break-out rooms for team activities. Overhead projector, flipcharts.

#### Topics

#### Introduction

#### **Understanding Facilitation**

- Context for facilitation
- Leading, coaching, and facilitating

#### Planning and Preparation

- Setting objectives and boundaries
- Environment and logistics

#### **Roles and Responsibilities**

- The facilitator's role
- The participants' roles and responsibilities

#### **Occasions for Facilitation**

- Group functions and goals
- Workshops, team meetings, and JAD sessions

#### Facilitation Skills and Techniques

- Verbal and non-verbal communication
- Managing disagreement constructively
- Handling difficult individuals
- Performing Facilitation
- Setting the scene
- Time keeping and record keeping
- Follow up

#### Conclusion

Format

C Presentation

Objectives

Written Exercises

♥ Interactive Activities

Plan and prepare for facilitation

Apply facilitation techniques

After completing this course, participants should be able to:

• Define the goals and constraints of different types of facilitation

• Define the roles of those involved in the facilitation process

Personal action plan

# Influencing Skills Workshop

#### 2 days

#### Description

In today's team-oriented and loosely structured organizations, it is often a challenge to obtain the agreement and commitment needed to work productively and meet deadlines. This interactive workshop introduces the skills and techniques of effective influencing and refines those skills through extended role playing activities.

#### Audience

IT professionals. This workshop can also be customized for other professionals.

Maximum number of participants: 12

#### Prerequisites

An understanding of communication skills is preferred, such as that provided by Arkoa's *Communication Skills for IT Professionals* workshop.

#### **Classroom Requirements**

Room set up as horseshoe and large enough for team activities, or with break-out rooms for team activities. Overhead projector, flipcharts.

#### Topics

#### Introduction

- Introduction
  - Housekeeping Introductions Objectives and Agenda
- Defining "influencing"
- Communication

- Body language *Reading Body Language Personal space Cultural Differences Using Body Language to Convey Messages Using Body Language to Create Rapport*
- Voice patterns
- Words and matching language Guidelines Matching Language Avoiding Overload
- Active listening skills
- Techniques for Effective Listening
- Representational systems
- Objectives in Communication Using Negatives to Positive Effect Good Communicators

#### Format

- Presentation
- Written Exercises
- Interactive Activities

#### Objectives

After completing this course, participants should be able to:

- Define "influencing" and identify factors that affect one's ability to influence others
- Identify the components of effective communication
- Describe "rapport" and the different levels at which it can be established
- Use effective negotiating techniques

- Logical levels and leverage
   Environment
  - Behavior

Levels of Rapport

- Capability
- Values and Beliefs

#### **Negotiating Skills**

- Types of negotiation
- Stages of negotiation
- Preparation Setting the Scene Bargaining Close
- Ground rules for negotiation
  - Follow-through
- Resolving conflict
- Using questions

#### Unconscious Thought Patterns

- Identifying thought patterns
- How they impact on communication

#### Course Review

# Interviewing Skills for IT Professionals

#### 2 days

#### Description

This interactive workshop provides an opportunity to learn the different phases of interviewing: preparation and planning, conducting an interview, and follow-up. It also provides role-playing exercises that enable participants to apply and refine their interviewing skills. NOTE: This course does not cover recruitment interviewing. See Arkoa's *Recruitment Interviewing Workshop* for this topic.

#### Audience

IT professionals who gather and assess information. The workshop can be customized for other professionals.

Maximum number of participants: 12

#### Prerequisites

No prerequisites.

#### **Classroom Requirements**

Room set up as horseshoe and large enough for team activities, or with break-out rooms for team activities. Overhead projector, flipcharts.

#### Topics

Professional Skills

Vorkshops

#### Introduction

**Positioning Interviewing Skills** 

#### Planning and Preparation

- Planning
- Collecting information
- Reviewing charts
- Preparing to interview
- Compiling questions
- Reviewing questions

#### Conducting the Interview

- Interviewing skills
- Preparing to interview
- Conducting the interview
- Reviewing the interview

#### Following the Interview

- Follow-up actions
- Evaluating information
- Cross-checking information

#### Conclusion

Post-workshop assessment

#### Format

- Presentation
- Written Exercises
- ♥ Interactive Activities

#### Objectives

After completing this course, participants should be able to:

- Evaluate the importance of interview planning and preparation
- Describe techniques for effective information gathering
- Design questions to facilitate the interview process
- Develop interview goals and define outcomes
- Structure and conduct interviews
- Identify interpersonal factors that can contribute to successful interviews

# Managing Change Within the IT Organization

#### 2 days

#### Description

This interactive workshop describes techniques for facilitating the change process. It helps to modify long-standing IT (information technology) organizational paradigms, examines types of change, and allows participants to explore the behaviors and feelings of those experiencing change. Roleplaying activities assist participants in becoming agents for change.

#### Audience

IT managers and professionals at all levels.

Maximum number of participants: 12

#### Prerequisites

Participants who are unfamiliar with current technology trends should attend the *Multi-Tier Client/Server Technical Overview*.

#### **Classroom Requirements**

Room set up as horseshoe and large enough for team activities, or with break-out rooms for team activities. Overhead projector, flipcharts.

#### Topics

#### Introduction

#### **Drivers of Change**

 Causes of change Business, organizational, and technology drivers Identify significant changes Identify drivers of change

#### Types of Change

- Scale and dimensions of change
- Impact of change
- Time frames for change

#### Technology, Organizational, and Process Models

- Technology models
- Organizational models
- Process models

#### Format

- Presentation
- Written Exercises
- Interactive Activities

#### Objectives

After completing this course, participants should be able to:

- Identify the causes of change in their companies
- Describe types of change
- Describe alternative organizational, process, and technology models
- Develop plans for person and work group change
- Handle resistance to change

#### Planning and Managing Change

- Planning for change
- Managing change

#### Resistance to Change

- Group and individual dynamics
- Techniques for facilitating change
- Conclusion

# **Negotiating Skills for IT Professionals**

#### 2 days

#### Description

Information technology (IT) professionals are more productive when they can negotiate effectively with customers, business users and managers, vendors, and fellow team members. This workshop explains how to use effective negotiating techniques. It describes how to plan for a negotiation, define goals and desired outcomes, prepare ideal, realistic and fallback positions, use bargaining techniques, and deal with difficult individuals and situations. The workshop also provides opportunities for refining negotiating skills through role-plays and discussions. Each topic is reinforced by exercises and activities.

#### Audience

IT professionals involved in both formal and informal negotiations. Maximum number of participants: 12

#### Prerequisites

Preferably, participants should have attended Arkoa's *Communication Skills for IT Professionals* workshop.

#### **Classroom Requirements**

Room set up as horseshoe and large enough for team activities, or with break-out rooms for team activities. Overhead projector, flipcharts.

#### Format

- Presentation
- Written Exercises
- ♥ Interactive Activities

#### Objectives

After completing this course, participants should be able to:

Course # 12-0300

- Define and set negotiating goals
- Describe how to prepare for negotiation
- Determine negotiating positions
- Describe effective negotiating techniques
- Outline strategies and tactics for handling difficult situations

#### Topics

#### Introduction

#### Phases of Negotiation: Preparation

- Context for negotiation
- Why negotiation is successful or unsuccessful
- Types of negotiation
- Preparation for negotiation

#### Phases of Negotiation: The Process

The negotiation process

#### **Negotiating Skills**

- Essential negotiating skills
- Preparing effective questions
- Using effective questions in negotiating

#### Follow-Through

- Following through a negotiation
- Follow-up action
- Making an action plan
- Reviewing the results of the action plan

#### **Negotiation Postures**

- Negotiating types
- Preparing postures and tactics
- Using postures and tactics
- · Reviewing the results of using postures and tactics

#### Conclusion

Personal action plan

20

# **Presentation Skills Workshop**

#### 2 days

#### Description

Good presentation skills project a professional image and assist in influencing decisions, presenting results, and providing information. This interactive workshop enables professionals to develop and refine their presentation techniques. Case study scenarios are used in exercises and can be customized. The workshop provides both a conceptual overview and team and individual exercises for refining presentation skills, with individual feedback and coaching.

#### Audience

No prerequisites.

#### Prerequisites

Anyone who would like to make more effective presentations.

Maximum number of participants: 12

#### **Classroom Requirements**

Room set up as horseshoe and large enough for team activities, or with break-out rooms for team activities. Overhead projector, flipcharts.

#### Topics

#### Introduction

#### **Planning the Presentation**

- Presentation goals
- Audience profile
- Establishing objectives
- Structuring the presentation
- Team perspective

#### **Developing the Presentation**

- Creating the content
- Designing the presentation
- Visual aids
- Handouts

#### Format

- Presentation
- Written Exercises
- Interactive Activities

#### Objectives

After completing this course, participants should be able to:

- Plan a presentation
- Define the content of a presentation
- Create a presentation outline
- Develop a presentation
- Recognize and apply effective presentation techniques
- Identify which presentation skills need continued development

#### **Delivering the Presentation**

- Presentation basics
- Nonverbal communication
- Using your voice
- Techniques for building confidence

#### Workshop Review

# Problem Solving and Decision Making Workshop

#### 2 days

#### Description

Success in today's rapidly changing and competitive environment depends upon being able to sort and analyze information, rapidly assess a situation, and come up with correct and timely solutions and decisions. However, today's professionals are often overwhelmed by a mass of information from many sources. This workshop explains how to distinguish between important and less important inputs and how to apply effective techniques to the processes of problem solving and decision making. The workshop also includes exercises for establishing and reinforcing these techniques.

#### Audience

Persons in any jobs or roles in which problem solving and decision making are important.

Maximum number of participants: 12

#### Prerequisites

An understanding of communication skills is desirable such as that provided by Arkoa's *Communication Skills for IT Professionals* workshop.

#### **Classroom Requirements**

Room set up as horseshoe and large enough for team activities, or with break-out rooms for team activities. Overhead projector, flipcharts.

#### Topics

#### Introduction

#### Drivers for Effective Problem Solving and Decision Making

- Motivating factors
   Business drivers
   Complex technologies
- Ways of working in IT Teams and methods

#### The Problem-Solving and Decision-Making Process

Problem resolution

#### Types of Problems and Solutions

- Problem identification
- Causes of problems
  - Causes versus symptoms Identifying the causes

#### Managing Information Inputs

- Gathering information
- Organizing and analyzing information

#### Creative Techniques for Problem Solving and Decision Making

 Creative techniques Techniques for creative problem solving Brainstorming Sorting ideas

After completing this course, participants should be able to:

ferences between symptoms and causes

Describe the problem-solving and decision-making life cycles

• Describe how to gather, organize, and analyze information inputs

Apply a variety of problem-solving and decision-making techniques

Apply appropriate techniques to select a solution and make a decision

Recognize the drivers for effective problem solving and decision making

· Identify different types of problems and solutions and recognize the dif-

#### **Selecting Solutions and Making Decisions**

- Selection techniques
- Presenting a solution or decision
   Obtaining commitment and agreement

#### Conclusion

Format

C Presentation

Objectives

Written Exercises

♥ Interactive Activities

Case Study Scenarios

# **Recruitment Interviewing Workshop**

#### 1 day

#### Description

This interactive workshop guides participants through the recruitment life cycle, from defining skill profiles to making a job offer. It demonstrates how to screen resumes, how to use effective interviewing techniques and how to assess candidates. It also discusses legal constraints. The workshop develops skills in individual and team activities and provides opportunities to refine interviewing skills in role-play exercises.

#### Audience

This workshop is for anyone involved in recruitment of IT (information technology) professionals. It can be customized for recruitment of other staff.

Maximum number of participants: 12

#### Prerequisites

No prerequisites.

#### **Classroom Requirements**

Room set up as horseshoe and large enough for team activities, or with break-out rooms for team activities. Overhead projector, flipcharts.

#### Topics

#### Introduction

#### Interview Requirements

- Identifying requirements
- Gathering and verifying information
- Results of ineffective interviews
- Improved interviewing skills

#### **Planning and Preparation**

- Define job requirements
- Job profile/rating system
- Finalize interviewing plan
- Types of interview
- Structure and format selection
- Short-listing applicants
- Preparing to interview
- Strategy
- Framing questions

#### Format

Presentation

#### Objectives

After completing this course, participants should be able to:

- Develop job profiles for recruitment purposes
- Screen resumes, using job profiles
- Plan and prepare recruitment interviews
- Design questions for interviews
- Structure and conduct interviews
- Document the results of interviews
- Design and use an analysis matrix to rate candidates
- Recognize legal constraints in recruitment

#### Conducting the Interview

- Interview Skills
- Using questions
- Assessing the reliability of responses
- Note-taking
- Tact and responsivity
- Telephone interviewing
- Reviewing the interview

#### Following the Interview

- Reviewing and evaluating interview results
- Selecting a candidate
- Making and closing an offer
- Conclusion

# Team Building for IT Professionals

#### 2 days

#### Description

Today, most organizations operate with formal or informal teams. High performance, successful teams must be planned, developed, and nurtured. This workshop describes how selecting team members, setting shared goals, defining team roles, and balancing work preferences can impact on team performance. It also provides opportunities for applying and refining skills through team role-playing.

#### Audience

IT (information technology) and business professionals involved in building, managing, or participating in teams.

Maximum number of participants: 12

#### Prerequisites

No prerequisites.

#### **Classroom Requirements**

Room set up as horseshoe and large enough for team activities, or with break-out rooms for team activities. Overhead projector, flipcharts.

#### Format

- Presentation
- Written Exercises
- ♥ Interactive Activities

#### Objectives

After completing this course, participants should be able to:

- Define the characteristics and responsibilities of a team
- Describe the factors that contribute to building a successful team through each phase of team development
- Identify the task and team roles and responsibilities in different types of teams
- Describe techniques to facilitate solving different types of problems in teams
- Describe techniques to measure and maintain a high level of team performance

#### Topics

٠

#### Introduction

#### Team Definitions

- Team definitions
  - Team characteristics Groups and teams Different types of teams Success criteria for different types of teams

#### **Building a Team**

- Team formation
   Team development stages
   Team development characteristics
- Successful teams
- Team objectives

#### Team Roles and Responsibilities

- Task-related roles and responsibilities
- Team-related roles and responsibilities

#### **Problem Solving**

- Types of problems and solution
- Problem-solving techniques

#### Team Maintenance

- Measuring team performance
- Maintaining team performance
- Managing disagreement and conflict

#### Conclusion

Personal action plan
# Train the Technical Trainer Workshop

#### 2 days

#### Description

Many highly skilled technical professionals feel uncomfortable in a teaching role because they lack instructional skills and techniques. This workshop develops and refines technical training skills through active involvement in a variety of exercises and role-plays. The workshop is also useful for educators who have some teaching experience but need to refine their skills or are new to technical training of adults.

#### Audience

Technical professionals and educators who deliver technical training. Maximum number of participants: 12

#### Prerequisites

An understanding of communication skills is desirable such as that provided by Arkoa's *Communication Skills for IT Professionals* workshop.

#### **Classroom Requirements**

Room set up as horseshoe and large enough for team activities. Overhead projector, flipcharts, video camera and VCR.

#### Topics

#### Introduction

#### **Training and the Learning Process**

- Learning process and styles
- Adult learners

#### Assessment Methods

- Assessment techniques
- Analyze the performance issue
- Is training the solution?

#### **Training Development**

- The development process
- Writing goals and objectives
- Developing the course
- Training approaches and materials
- Evaluation strategies

# Format

- Presentation
- Written Exercises
- Interactive Activities

# Objectives

After completing this course, participants should be able to:

- Identify the different ways in which people learn and retain information
- Write measurable learning objectives
- Prepare lesson plans and utilize training aids
- Apply and combine different teaching techniques
- Plan and deliver an effective presentation
- Measure and evaluate training effectiveness
- Identify and apply facilitation techniques

#### **Training Delivery and Facilitation Skills**

- The role of the trainer
- Training environment and tools
- Managing the delivery
- Training delivery skills

#### **Evaluation Methods**

- Levels of evaluation
- Evaluation tools
- Principles of feedback

#### **Presentation Practice**

- Designing the opening, body, and closing of a presentation
- Team presentation perspective

# **Business Requirements Workshop**

#### 2 days

#### Description

This interactive workshop provides an opportunity to develop and refine skills for defining business, user, and system requirements. The workshop reviews techniques for gathering, verifying, and documenting business requirements. It discusses prioritization, explains the importance of tracing requirements, and provides extended exercises, using case study scenarios or actual projects.

#### Audience

IT professionals who are involved in the analysis, design, and development of systems.

Maximum number of participants: 12

#### Prerequisites

Participants should be familiar with the basics of analysis and design.

#### **Classroom Requirements**

Room set up as horseshoe and large enough for team activities, or with break-out rooms for team activities.

Overhead projector, flipcharts.

#### Topics

# Introduction

#### **Business Drivers and Goals**

- Customers, competition, and change
- Recognizing industry and company drivers

#### Requirements

- Stakeholders and requirements
- Case study scenario

#### Information Gathering

- Problems in information gathering
- Organizing and recording information
- Techniques for communicating information
- Information sources
   Customer documents and environment
   Completeness and accuracy

#### **Documentation Requirements**

- Organizing and recording information
   *Requirements document*
- Interpreting requirements

#### Format

- Presentation
- Written Exercises
- ♥ Interactive Activities

#### Objectives

After completing this course, participants should be able to:

- Recognize the implications of business drivers and goals for defining business requirements
- · Describe typical stakeholders and their views of business requirements
- Define categories of requirements and business requirements within the categories
- Apply techniques that facilitate information gathering and verification
- Document requirements in an organized manner
- Apply techniques that prioritize requirements
- Describe the process for translating requirements into specifications
- Prepare a Requirements Document

# Verifying Requirements

- Verification and validation Identifying requirements for validation Verification tactics
- Negotiation
  - Negotiating tactics Negotiating with subject matter experts

#### **Prioritizing Requirements**

- Analysis techniques
- Individual and group prioritization

#### Measuring and Testing Requirements

- Testing
- Input to testing

#### Specifications

- Functional specifications Specification process Levels of detail
- Conclusion

# **Client/Server GUI Design Workshop**

#### 1 day

#### Description

Good GUI design is essential for client/server applications, with the interface reflecting user requirements. It is important for prototyping, usability testing, and user acceptance. This one-day workshop explores the process of analyzing user requirements, applying good design principles, and following standards to design a GUI that will enhance user productivity.

#### Audience

This course is intended for GUI developers. Depending upon the participants' experience with GUI tools, the exercises may all be written or some may be on the computer using a tool such as Visual Basic, Visual C++, or PowerBuilder.

#### Prerequisites

Basic programming skills and an understanding of a window-based user interface are required.

#### **Classroom Requirements**

Room set up as horseshoe and large enough for team activities, or with break-out rooms for team activities. Personal computers are helpful but not required.

# Topics

#### Introduction

- Housekeeping and Administration
- Introductions
- Agenda
- Setting a Context for GUI Design
- Driving Forces
- Types of Computer Interfaces
- Learning and Working Styles
- Good Design
- GUI Design
- GUI Design Principles
- GUI Elements
- ♦ GUI Design Process

#### Analysis

- Analysis Process Alternatives
- Identify Objectives
- Assess User Profiles
- Assess Workflow
- Identify Database Interactions
- Assess Work Environment
- Define Test Criteria
- Produce Requirements Document

# Format

- Presentation
- 🖪 Hands-on
- Written Exercises

# Objectives

After completing this course, participants should be able to:

- Understand the context for GUI design, including the role of human behavior in design
- Describe the GUI design process and principles
- Analyze user requirements for the GUI
- Translate requirements into effective navigation design
- Make appropriate choices for effective screen design
- Recognize the different levels of help and guidance
- Assist in setting standards and follow GUI style and design standards

#### **Navigation Design**

- Navigation
- Define Conceptual Model
- Define Task Flow
- Define Dialog
- Define Defaults and Constraints
- Define Shortcuts
- Navigation Considerations

#### Screen Design

- Types of Interaction
- Icons, Buttons, Toolbars
- Fonts, Colors, and Symbols
- Presenting Information

#### **Designing Guidance and Help**

- Errors
- ♦ Help
- Testing

#### Types of Testing

#### Setting and Following Standards

- Guidelines and Standards
- Windows Interface Standards

#### Conclusion

GUI Design Success Factors

#### Appendix A: Case Study

#### Appendix B: Client/Server glossary

ystems Development and Training

#### 1 day

#### Description

This course introduces the non-programmer to fundamental programming concepts. The course covers the basics of computers and computer programs. The course also covers the software development process and what is involved in professional software development.

#### Audience

Prospective programmers or anyone involved in the software development process.

#### Prerequisites

Although no prior experience is necessary, the course assumes that the students have a basic level of computer literacy.

#### Hardware and Software

Any version of Windows and latest version of preferred Web browser, text editor.

# Format

- Presentation
- \rm Hands-on
- Demonstration

#### Objectives

After completing this course, participants should be able to:

- Describe how a computer program works
- Define important programming terms
- Define a programming language and discuss several important programming languages
- Explain how data is used in programming
- Explain programming concepts such as variables, operators, boolean logic, statements, looping and flow control, events
- Discuss Web application programming
- Discuss object-oriented programming concepts
- Discuss the software development process
- Discuss debugging approaches

# Topics

# Programming Basics

- Computer Basics Main Computer Components Central Processing Unit Main Memory Mass Storage
- What is a Program? *Computer Programs Algorithms Machine Language Compiling Linking and Loading Interpreter File Naming*
- Executing Programs
   Program Execution
   Platform Independence
   Processes
   Threads
- Computer Languages

   Language Syntax
   Language Semantics
   Languages
   Web Page Script
   Language Syntax Fundamentals

# **Data and Operators**

- Data Types
  - Types of Data Character Data Working with Character Data

- Integer Data Floating Point Data Floating Point Advantages/Disadvantages Money Data Date Data
- Variables and Constants What is a Variable? Naming Variables Example Names Declaring Variables Assigning Values to Variables Literals and Constants Special Character Literals Arrays
- Operators
  - What Is an Operator? Arithmetic Operators Parenthesis Concatenation

# **Conditional Processing and Functions**

 Conditional Processing Conditions If Statement Boolean Logic Statement Blocks Switch Statement Iteration Terminating Loops Course # 02-0100

- Functions
  - What is a Function? User-Defined Functions Function Arguments Scope: Definition Scope: Analysis Scope: Effects Scope: Guidelines Function Return Values

#### Interacting with the User and Object Oriented Programming

 Interacting with the User Ways to Interact

Client/Server Processing Web Applications CGI What Can and Cannot be Done with Script Programming Web Pages Document Object Model Using the Document Object Model Document Object Model Details Event-Driven Programming Processing Forms

- Processing Buttons
   Handling Numbers
   Object-Oriented Programming
- Programming Paradigms What is Object-Oriented Programming? Classes and Objects Constructors and Destructors Creating JavaScript Objects Encapsulation Inheritance JavaScript Inheritance Instance Variables Instance Methods Class Variables Class Methods Class/Instance Methods and Variables Example: Complex Number Object Example: Using Complex Number Objects Built-In JavaScript Objects

#### **Object-Oriented Programming and Software Development**

 Object-Oriented Programming Programming Paradigms What is Object-Oriented Programming? Classes and Objects Constructors and Destructors Creating JavaScript Objects Encapsulation Inheritance JavaScript Inheritance Instance Variables Instance Methods Class Variable Class Method Class/Instance Methods and Variables Example: Complex Number Object Example: Using Complex Number Objects Built-In JavaScript Objects

 Software Development Software Life Cycle Development: Analysis Development: Design Development: Implementation Development: Testing Variations in Development Debugging: Compiled v. Interpreted Debugging: Alerts Debugging: Debug Flag Debugging: Try/Catch Debugging: Using Debuggers Debugging: Debuggers Documentation: Types Documentation: Comments Documentation: User

#### 1 day

#### Description

This course provides an overview of current testing and methods. It describes different types of testing, including user acceptance and system testing and how to develop plans to conduct those tests. Current industry practices and standards are discussed, and practical techniques for achieving compliance with those standards are explored.

#### Audience

Testing specialists and managers, product development managers, and programmers who need a technical introduction to writing test plans and managing the testing process.

#### Prerequisites

Familiarity with the software development lifecycle is required. Practical experience in testing and software development is recommended.

#### **Classroom Requirements**

Overhead projector, flipcharts

#### Topics

#### Introduction

#### Testing Overview

- Software quality assurance process
- Objectives of testing
- Testing stages

#### **Testing Techniques**

- White-box versus black-box testing
- Unit testing
- Integration testing
- System testing
- Regression testing

#### The Test Plan

- The test plan template
- Components and consideration
- Test deliverables
- QA project management
- Testing tasks and schedule

# Format

Presentation

# Objectives

After completing this course, participants should be able to:

- Identify industry standards and practices for test activities
- Describe what is needed to manage the testing process
- Define test case and test plan templates
- Write test plans that address the major categories of testing
- Select effective techniques for capturing test results and producing reports
- Define policies for managing the testing environment and testing procedures

#### **Test Cases**

- The testing matrix
- Types of test cases
- Creation of data sets
- Establishing baselines
- · Determining test inputs and outputs
- Test coverage assurance

#### **Test Execution**

- Capturing test results
- Controlling the test environment
- Updating test scripts
- The defect tracking cycle

#### **Code Migration**

- Production assurance
- Configuration management
- Backout procedures

#### Review

- Summaries
- Questions and answers



# **Object-Oriented Testing Methods**

#### 1 day

#### Description

This course provides a technical overview of current object-oriented testing practices and methods. The course covers a broad range of testing issues, from low-level unit test techniques to user acceptance test strategies and techniques. Current tools and methodologies are also surveyed.

#### Audience

Testing specialists, programmers, and product development managers who need a technical introduction to object-oriented testing methods.

#### Prerequisites

Familiarity with testing techniques and practices and with object-oriented technologies and languages is required. Object-oriented programming experience is recommended.

#### **Classroom Requirements**

Overhead projector, flipcharts.

#### Topics

#### Introduction

#### Introduction to Object-Oriented Testing

- Current state of object-oriented testing
- Issues unique to OO testing
- When and what should be tested
- Who should do the testing

# Use Case-Based System Testing

- Requirements defined through use cases
- Incremental systems testing
- Planning test coverage with operational profiles and risk analysis
- Generating test cases from scenarios

#### **Class Design for Testability**

- Using software contracts in class specifications
- Assertions
- Preconditions
- Postconditions
- Invariants
- State transition constraints
- Managing polymorphism

# Format

# Presentation

# Objectives

After completing this course, participants should be able to:

- Plan and use object-oriented testing techniques within the software development life cycle
- Select appropriate tools and techniques for testing object-oriented code and systems
- Apply design for testability in software designs
- Identify and select appropriate metrics to quantify the quality of code

#### **Component (Unit) Testing**

- Inputs to component testing
- Evaluating specifications and planning test coverage
- Testing strategy for components
- Test cases for methods
- State based testing
- Alternatives for test frameworks
- Incremental component testing
- Improving component level testing

#### Metrics

- Factors that affect quality
- Class design metrics
- Analyzing metrics

#### **Testing Tools**

- Test management tools
- Code and systems analysis tools
- Capture/playback tools

31

# **Client/Server Project Management Workshop**

#### 1 day

#### Description

This is an interactive workshop that reviews the processes involved in planning and managing client/server projects. It reviews different project management methods and techniques and identifies the roles and responsibilities of project team members. During the course of the workshop, participants develop project plans for sample client/server projects.

#### Audience

Project managers and IT professionals involved in a client/server development project, as a project manager, team or group leader, or individual project member.

#### Prerequisites

Attendance at Arkoa's *Multi-tier Client/Server Technical Overview* or familiarity with the client/server software development life cycle is necessary. It is also desirable to have an understanding of GUI design.

#### **Classroom Requirements**

Overhead projector, flipcharts.

#### Topics

#### Introduction

#### **Client/Server Development Life Cycle**

- Stages in software development
- Methods for software development
- Iterative design
- Incremental or parallel development

#### Project Management Review

- Project management scope
- Project management stages

# Preparation

- Evaluate project requests
- · Gathering and assessing project requirements
- Project preparation

#### The Team Perspective

- Building the team
- Effective team management

#### Planning

Project Management

- Detailed plans
- Work breakdown: phases and tasks
- Schedules
- Estimating
- Test plans
- Documentation plan
- Quality management plan
- Communication plan
- Training plan
- Change control

#### Format

- Presentation
- Written Exercises

# Objectives

- After completing this course, participants should be able to:
- Identify the phases in the client/server life cycle
- Describe the stages and activities in project management
- Prepare for projects
- Develop project plans
- Assign resources and build teams
- Manage projects
- · Identify potential risks and plan for contingencies
- Describe quality management
- Identify the skills and infrastructure needed to build client/server applications

# **Project Management**

- Administration
- Project tracking

# Managing Risks and Contingency Planning

- Risk impact on project
- Assessing project risks
- Assessing project failure risks
- Contingency planning

#### **Project Management Tools**

- Features
- Microsoft Project

#### **Quality Management**

- Quality assurance
- Conclusion
- Objectives
- Success factors

32

# **C** Programming

#### 5 days

# Description

This course provides programmers with a hands-on introduction to the C programming language. The course teaches both syntax and semantics of the C language. It also includes advice on good programming practices.

#### Audience

Systems and applications programmers who need a practical in-depth understanding of ANSI C.

#### Prerequisites

Participants should be familiar with one or more programming languages or have attended Arkoa's *Introduction to Programming* course. This course is not suitable for people without programming experience.

#### Hardware and Software

An ANSI C compiler running on any platform. The systems should be able to load a 3.5" DOS-formatted solution disk.

#### Topics

# An Introduction to C

- An Example
- The Compilation Process
- White Space and Comments
- Functions
- Constant Strings
- Libraries
- printf()
- Simple Statements
- Compound Statements
- Fundamental Types
- Defining Variables
- Identifiers
- Operators
  - Operators vs. Expressions
- The if Statement
- The while Statement
- The for Statement
- Character Handling
- The Preprocessor
- Standard Input/Output Library
- getchar() and putchar()

# **Data Types and Constants**

- Data Types
   Numeric and Character Types
   Sizes of Data Types
   Constant and Volatile Objects
- Constants
   Integer Constants
   Floating-point Constants
- Character Constants

# Functions

- What is a Function?
- Function Prototypes

- Format
- Presentation
- 😐 Hands-on

# Objectives

- After completing this course, participants should be able to:
- Use the syntax, semantics, and scoping of ANSI C
- Write ANSI C programs using many features of the language
- Use some of the ANSI C libraries
- Describe the C compilation process
- Write programs using dynamic data structures and recursion
- Recognize the advantages of multi-file programs

- ♦ The exit function
- Function Parameters
- Call By Value
- Return Values
- The void Type
- Writing Functions in Pre-ANSI C

# Declarations

- Declaring Variables
- ♦ Scope
  - Internal Variables Parameters to Functions External Variables Masking Scope of Functions
- ♦ Extent
  - Extent of Externals Automatic Variables Internal Static Variables Internal Static Variables – Example

# Initialization

#### Recursion

- What is Recursion?
- Why Recursion?
- Example

# **Evaluating Expressions**

- Truth Values
   Precedence
  - Associativity
- Table of Operators
- Operand Evaluation
- Expression Types and Conversions Compile-Time vs. Run-Time Precedence

C/C++ Programming

# **More Operators**

- Boolean Operators
   Order of Evaluation
- The Comma Operator
- The Conditional Operator

#### Assignment

- The Simple Assignment Operator
- Compound Assignment Operators
- Use of Assignment in Expressions

#### **Further Flow Control**

- The do-while Loop
- The Null Statement
  - A Caution About do . . . while ( )
- ♦ Continue
- Break
- Goto
- Switch

# Pointers

- Pointer Concepts
- Defining Pointers
- Pointer Operators
- Pointers As Function Parameters
- Pointers As Return Values

# Arrays

- Defining Arrays
- Using Arrays
- Initializing Arrays
- Multidimensional Arrays
- Strings
  - Processing Strings
- Arrays and Pointers
- Arrays as Function Parameters
- String Handling
- Library Function: gets()
- Some Library String Functions
- Pointers to Constants

# Pointers Revisited

- Pointer Arithmetic
- Incrementing And Decrementing Pointers
- String Handling
- Pointers To Functions
- Pointers to Functions Example
   Handling Complex Declarations
- Handling Complex Declarations
   Handling Complex Declarations Declaration Procedure
  - The Right Left Rule

# Structures

- Structure Declaration Name Spaces
- ♦ Structure Members
- Structure Initialization
- Operations on Structures

# Building New Data Types

- Data Abstraction typedef
- Towards Complexity One Step at a Time
- Typedef and Structures

# Dynamic Data Structures

- Self Referential Structures
- Linked Lists
- Dynamic Memory Allocation

# Multi File Programs

- Issues in Multi File Programs
- ◆ Variables extern and static

# Bit Twiddling

- Introduction
- Bit Twiddling Operations Setting and Unsetting Bits Bit Shifting
- Portable Bit Manipulation
- Printing Bit Representations
- Example Bit Manipulation

# **Additional Data Types**

- Enumerated Data Types
- Bit Fields
- Unions

# The Preprocessor

- Introduction
- ♦ #define
- ♦ #ifdef and #ifndef
- ♦ #if

# The Standard Libraries STDIO And Others

- The Concept of a FILE
- Character Handling Routines
- Line and Record Handling Routines
- Formatting Routines
- Positioning Routines
- The ANSI Libraries

# **Program Arguments**

- Invoking Programs
- Parameters to main()
- Processing argc and argv

# **ANSI C Coding Guidelines**

- Introduction
- Programming vs. Program Design Specific Programming Features To Avoid Suggestions Related To Overall Design
- Specific Coding Guidelines
  - Comments
  - Coding For Readability
  - Indentation
  - Brackets
  - Well-Chosen Identifier Names
  - Syntactical Forms To Be Avoided

# **Advanced C Programming**

#### 5 days

#### Description

This course gives C programmers practical experience applying advanced C programming techniques. The course is operating system independent.

#### Audience

Systems and applications programmers and software designers who will be using C for software development and need to understand every aspect of the C programming language.

#### Prerequisites

Participants must have a working knowledge of C programming language, or have attended Akoa's *C Programming* course.

#### Hardware and Software

An ANSI C compiler running on any platform. The systems should be able to load a 3.5" DOS-formatted solution disk.

# Format

- Presentation
- 🖪 Hands-on

# Objectives

After completing this course, participants should be able to:

- Use the enhancements offered by ANSI C
- Use all the functions in the ANSI library
- Write sophisticated applications using multiple source files and separate compilation
- Implement algorithms involving complex data structures
- Describe all of the techniques and facilities for error trapping and error recovery
- Identify potential barriers to portability

# Topics

# Debugging

- Debugging tools
- Code view Microsoft C
- sdb and dbx

# Working with Multiple Files

- Compilation and linking
- Scope rules revisited
- Include files

# MAKE

- Basic features
- Makefile macros
- The Makefile description file
- Adding new rules

# The Standard C Library

- Character handling
- Mathematical functions
- Standard I/O
- General utilities
- String handling
- Block memory functions
- Date and time functions
- Regular expressions

# **Data Abstraction and Advanced Algorithms**

- Dynamic data structures
- Self-referential structures
- Linked lists
- Doubly linked lists queues
- Binary trees
- Balanced (AVL) binary trees
- Flattening the search

# **Behavior of Complete Programs**

- The invocation environment
- Command line parser- getopt
- Interfacing to whole programs
- Program termination
- Exception handling
- Controlled backtracking

#### 5 days

#### Description

This course provides a practical hands-on introduction to the C++ programming language. Participants will learn the central concepts of the C++ language, with emphasis on the use of object-oriented techniques in writing robust code. Some advanced features are included.

#### Audience

Systems and applications programmers with substantial experience in the design and implementation of major software projects.

#### Prerequisites

Participants must be fluent in ANSI C and have an understanding of objectoriented concepts

#### Hardware and Software

An ANSI C++ development environment, including a compiler with template support, is required. A compiler that supports exception handling is recommended. The system must be able to load text files from a 3.5" DOSformatted solution disk.

#### Topics

#### Why C++?

- ♦ C vs. C++
- Abstract data types
- C++ as a better C
- I/O services
- Standard template library
- Standard compliance

#### Functions

- Function overloading
- Default parameter values
- Variable numbers of arguments
- Ambiguity
- Inline functions
- References
- The const type-qualifier

#### Classes

- Class definition
- Instantiating and using classes
- Member functions
- Classes and encapsulation
- Constructors and destructors
- Constructors and conversions

# Storage Management

- Memory allocation
- Dynamic allocation
- Problems with dynamic allocation

#### **Operator Overloading**

- Simple operator overloading
- Return values of operator functions
- Predefined operator functions
- Conversions
- Member vs. non-member functions
- Overloading I/O operators

# Format

- Presentation
- 🖪 Hands-on

# Objectives

- After completing this course, participants should be able to:
- Describe object-oriented programming techniques
- Use the syntax and semantics of the C++ programming language
- Create new classes
- Write an object-oriented program in C++
- Describe the process of data abstraction
- Create new data types using inheritance
- Use C++ class libraries
- Implement exception handling
- Write template functions and classes
- Friend functions
- The effect of conversions
- Restrictions on overloading
- The [] operator

#### Initialization

- Initialization vs. assignment
- Constructors
- Arrays of objects
- Branching past initialization
- Constant class members
- Static class members

#### Inheritance

- Derivation
- Overriding names
- Constructor and destructor calls
- Inheritance at work
- Conversions
- · Features not inherited
- Protected members
- When to use inheritance

#### Polymorphism

- Polymorphism
- Abstract classes
  - New casting operators

#### I/O in C++ Programs

- Standard streams
- Insertion and extraction operators
- Manipulators
- Unformatted input and output
- · File input and output
- Stream states
- Problems with object I/O

# **Exception Handling**

- Exception handling concepts
- Exception handling terminology
- Uncaught exceptions
- Inheritance and exceptions
- When in an exception handler
- Function declarations

# Templates

- Template functions
- Template classes
- Argument types
- Standard Template Library
- Implementation issues
- Vector of strings
- Maps

# Appendix: Review of C

#### 5 days

#### Description

This course makes C++ programmers more productive. It focuses on effective use of the language. This course covers advanced language features, describes useful implementation techniques, and reviews object-oriented design guidelines.

#### Audience

Systems and applications programmers who need to use C++ to design and write new software.

#### Prerequisites

Participants must have attended an introductory C++ course to the level of Arkoa's C++ *Programming*, and should have at least three to six months of experience using C++.

#### Hardware and Software

A C++ development environment, including a compiler with template support, is required. A compiler that supports exception handling is recommended. The system should be able to load a 3.5" DOS-formatted solution disk.

#### Topics

#### Experienced Use of C++

- Using constants and constant pointers
- References, parameters, and variables
- Inline functions and encapsulation
- Templates and initialization

#### **User-Defined Data Types**

- Defining robust classes
- Constructors and initialization
- Dynamic storage within a class
- Type conversion and array types
- Stand-in classes

#### Generic Data Types

- Collection classes
- Data type vs. data structure
- Class templates
- Names and instantiation
- Specialization

# Format

- Presentation
- 🗳 Hands-on

#### Objectives

After completing this course, participants should be able to:

- Identify all the major features of the C++ language, including templates and exceptions
- Discuss issues of inheritance
- Use a powerful set of C++ techniques and idioms for solving problems
- Apply object-oriented design concepts to C++ development

#### **Class Relationships**

- ♦ Inheritance
- Virtual functions, polymorphism, and abstract base classes
- Virtual destructors and base classes
- Constructors

#### Memory Management

- Dynamic object creation
- Vector allocation
- Sharing memory among objects
- Reference counting and cycles
- Swappable objects
- Smart pointers

#### **Exception Handling**

- Error-handling strategies
- Throwing exceptions
- Inheritance and exceptions

#### Mixing C and C++

- ♦ C/C++ source and link incompatibility
- ♦ Calling C++ code from C

C/C++ Programming

# C++ Standard Template Library (STL) Programming

#### 3 days

#### Description

This course provides a practical hands-on introduction to the C++ Standard Template Library (STL). Participants will learn the central concepts of the STL, with emphasis on its use to create reliable, scalable applications. Many advanced features are discussed.

#### Audience

Systems and applications programmers with substantial experience in the design and implementation of major software projects and an interest in using the STL.

#### Prerequisites

Participants must be fluent in C++ and have an understanding of object-oriented concepts

# Hardware and Software

An ANSI C++ development environment with the STL and a compiler with template support, is required. An Internet connection is recommended. The systems must be able to load text files from a 3.5" DOS-formatted solution disk.

# Format

- Presentation
- 🖪 Hands-on

# Objectives

After completing this course, participants should be able to:

- Use the STL to solve common programming problems
- Explain the STL sequence and associative container classes and adapters, and list their advantages and relationships
- Describe iterators, differentiating the different classes and explaining ranges and ordering properties
- Use the STL generic algorithms to solve practical programming problems
- Develop and use function objects and adapters
- Design and implement efficient, reliable, readable, and maintainable programs using the STL.
- Explain the STL performance guarantees
- Use the STL reference material
- Discuss methods to extend STL functionality

# Topics

# STL Introduction

- STL Basic Concepts STL and the Standard C Library
  - Container Basics and Overview Iterator Basics Algorithm Basics The STL Namespace
- C++ Template Review Using Inline Functions (Instead of #define Macros) Function Overloading Function Templates STL and Templates

# Sequence Containers and Their Algorithms

- Sequence Containers Overview Concepts and Requirements Vectors and their Algorithms Lists and their Algorithms Deques and their Algorithms Sequence Container Summary
- Container Adapters Overview Stacks Queues Priority Queues
- Interlude: STL Additional Information Web Sites Other References

#### Iterators, Allocators, and Algorithms

- Iterators and their Conventions
   Overview
   Iterator Requirements
   Using Iterators
- Mutable Iterators
   Iterator Orderings
   Iterator Adapters
- Algorithms, Ranges, and their Conventions Ranges Using Ranges with Algorithms
  - Stream Iterators
- Allocators
  - Allocator Rationale Allocator Types and their Usage

# Associative Containers and their Algorithms

- ♦ Associative Container Overview
  - Set, Multiset, Map and Multimap Introduction Keys and Data
  - Ordering
  - Object Definitions for Set and Map Examples
- Sets and Multisets
  - Set and Multiset Overview
    - Sets
    - Multisets Set and Multiset Algorithms
- Using Maps and Multimaps Map and Multimap Overview

Maps Multimaps Associative Container Performance and Data Structures

# Generic Algorithms and Function Objects

- Function Objects and their Use Function Object Concepts Writing Function Objects Unary and Binary Function Templates Built-in Function Objects
- STL Generic Algorithms
   Generic Algorithm Concepts
   Non-Mutating Sequence Algorithms
   Mutating Sequence Algorithms
   Sorting and Searching
   Numeric Algorithms
   Miscellaneous Algorithms
- Function Adapters
  - Function Adapter Concepts Negaters Binders Pointer to Function Adapters Composition Adapters

# Visual C++ (6.0) and the MFC Library

#### 5 days

#### Description

This course provides participants with a hands-on introduction to building Windows applications. It makes full use of Visual C++ (6.0) tools. The exercises provide solutions to common problems. The course covers the MFC library, the Visual C++ Document-View-Frame model for building standard Windows applications, the MDI (Multiple Document Interface), and creation of OLE-compliant compound document files.

#### Audience

Programmers who will be building Windows applications using MFC.

#### Prerequisites

Participants must have practical experience programming in C++. Experience with the Windows SDK is helpful but not required.

#### Hardware and Software

Microsoft Visual C++ v6.0, running on Windows 95/98/ME/NT/2000. Pentium systems are recommended to shorten compile times. The systems should read a 3.5" solution disk.

# Topics

# Windows Application Management

- Application Management Under Windows
- Windows API Programming
- MFC Windows Class Hierarchy

# Introduction to the Tool

- AppWizard Overview
- The Simple Application
- Project | Add to Project | New
- Project | Add to Project | Files
- Build | Build, Build | Rebuild All, and Build | Execute
- File | Open Workspace and File | Close Workspace
- AppWizard
- Executing the Simple Application

# Other Wizards and Debugging

- ClassWizard
- The Message Map
- Resource Editors
- The Debugger
- Using The Integrated Debugger
- Setting Breakpoints
- Edit and Continue
- Project Settings

# Managing Resources

- Resource Defined
- Resource Files, Compilation, and Binding
- Resource View
- Inserting New Resources

# Menus

- The Menu Editor
- Menus and the Keyboard Interface

# Format

- Presentation
- 🖪 Hands-on

# Objectives

After completing this course, participants should be able to:

- Use the AppWizard and other Visual C++ tools
- · Build user interfaces using views, menus, dialog boxes, and controls
- Describe the Windows event-driven application model as seen through the Visual C++ environment
- Use ActiveX controls in Windows applications
- Use ODBC for database access with Visual C++
- Describe MFC's document/view architecture
- Design advanced windows

# **Events and Handlers**

- Events and Handlers
- The ClassWizard
- Handling Options and Implications
- Handling Menu Events

# **Dialog Boxes**

- Dialog Boxes
- The Dialog Editor
- Creating a Dialog Class
- Displaying the New Dialog Box
- Automatic Data Exchange
- Modeless Dialogs
- Common Dialogs

# **Basic Controls**

- Accessing Controls from the Dialog Class
- Static Text Controls
- Edit Box Controls
- Push Button Controls
- Static Text Example
- Check Box Controls
- Radio Button Controls
- Group Box Controls
- Check Box and Radio Button Example
- Picture Box Controls

# List Box Controls

- List Box Controls
- Combo Boxes

#### Windows Common Controls

- Progress Bar Control
- Slider Control
- Spin Control
- List Control
- Tree View Control

#### Course # 08-0700

# Using ActiveX Controls

- ActiveX Components
- Installing ActiveX Components
- ◆ Adding ActiveX Components to a Project
- ♦ MSFlexGrid Control

# **Database Access with ODBC**

- Rationale for Using Open Database Connectivity
- ♦ MFC Classes for ODBC
- ◆ ODBC Configuration For Windows
- ◆ MFC ODBC Example

#### **Documents and Views**

- ◆ The Document-View Architecture
- ♦ Multiple Views of Data
- Finding a View's Document and Vice Versa
- Processing User Input
- Painting in Windows

#### Advanced Window Design

- Interacting with the Framework
- ♦ SDI and MDI Applications
- ♦ MFC Architecture
- Splitter Windows
- Window Design Options

# Data Warehouse Technical Overview

#### 1 day

#### Description

Data warehouses are able to turn raw data into business intelligence that supports effective decision making and provides a competitive advantage. However, designing and building a data warehouse requires a new way of thinking and substantial investment. This overview provides a high-level introduction to data warehouse alternatives and issues.

#### Audience

Managers and professionals evaluating or implementing data warehouse technology.

#### Prerequisites

A basic understanding of relational databases is preferred.

#### **Classroom Requirements**

Overhead projector, whiteboard, flipcharts.

#### Topics

# Introduction

#### **Driving Forces**

- Business pressures
- Technology drivers

#### **Data Warehouse Definitions**

- Why consider a data warehouse?
- What is a data warehouse?
- Benefits and risks ٠

# **Data Warehouse Capabilities and Applications**

- Data storage
- Operational data
- Informational systems
- Analytic systems ۲
- Tools and their use ٠

# Format

- Presentation
- Written Exercises

# **Objectives**

After completing this course, participants should be able to:

- Describe the business benefits of a data warehouse
- Describe different types of data warehouses
- Identify the front- and back-end components of a data warehouse ٠ · Identify the critical success factors in designing and building a data
- warehouse

# **Data Warehouse Architecture Components**

- Data warehouse elements
- Middleware
- Database engines
- Applications
- Datamarts

# **Data Warehouse Design and Development**

- Defining requirements
- Modeling data
- Cleansing data
- Performing extraction, transformation, and loading
- Maintaining the data warehouse

# Conclusion

Critical success factors

# Introduction to SQL

# 2 days

# Description

This course introduces the most popular database query language, Structured Query Language (SQL). The ANSI syntax for SQL is illustrated through examples. Oracle and SQL Server extensions are also discussed. The basics of querying, modifying, and adding data are practiced through activities.

# Audience

Application developers, business managers, and users who need to develop SQL statements.

# Prerequisites

Familiarity with basic operating system commands is required.

# Hardware and Software

Either Oracle8 or Microsoft SQL Server 7 is recommended. Personal Oracle or Sybase SQL Anywhere can also be used.

# Topics

# Introduction to SQL

- The relational model
- Why SQL?
- Queries and results
- Inputting SQL commands
- The Cars database example

# **Single Table Queries**

- Projecting data with SELECT Writing queries with SELECT Selecting distinct data Modifying results with expressions Controlling column headings Incorporating functions Working with NULL data Grouping related rows Using COMPUTE
- Restricting data with SELECT *Comparing expressions Comparisons with NULL data Comparisons with LIKE, BETWEEN, and IN Conjunctive operators Restricting aggregate result sets with HAVING*
- Sorting results with ORDER BY

# Format

- Presentation
- 😐 Hands-on
- Written Exercises

# Objectives

After completing this course, participants should be able to:

- Describe the history and use of SQL
- Query selected data
- Format result sets
- Evaluate queried data using functions and expressions
- Join data from multiple tables
- Develop queries to answer business questions
- Insert, update, and delete data

# **Multiple Table Queries**

- Joining data
  - Joining data from multiple tables Using table aliases Join examples
- Advanced joins and multiple-table operations Self-join statements Outer joins Set operations Subqueries

# **Modifying Data**

- The INSERT statement
- Using SELECT INTO in SQL Server
- The DELETE statement
- The UPDATE statement

# Logical Data Modeling

#### 3 days

#### Description

This course explains how to perform logical data modeling (LDM) and design a database suitable for the type of data it will hold. Participants will learn how to transform business information requirements into a logical data model and initial database design. This course includes a series of exercises in basic and advanced design.

#### Audience

System analysts/designers, application developers, and database administrators.

#### Prerequisites

A general introduction to databases is recommended but not required. This course should be followed by Arkoa's Relational Database Design course. SQL, Sybase, and Oracle courses are also available.

# Format

- Presentation
- Written Exercises

# **Objectives**

After completing this course, participants should be able to:

- Identify and define business information requirements
- Recognize the principles behind data architectures and corporate data ٠ models
- Select an appropriate approach to data modeling
- Create a logical data model
- Describe the transition from a logical data model to a physical relational ٠ database design
- Use advanced techniques to refine the data model

# Topics

# Introduction

- Definition, benefits, and risks
- Corporate vs. project data modeling
- Logical vs. physical data modeling
- Building the LDM

# Approaches to Logical Data Modeling

- Top-down, bottom-up, and view integration approaches
- Selecting the right approach
- **Entity-Relationship Modeling**
- Why E-R modeling?
- CASE tools
- E-R modeling concepts
- Relationships
  - Connectivity and cardinality Optionality
  - Named
  - Many-to-many
- Validating E-R models

# Transactions vs. Decisions

- DSS vs. OLTP
- Data warehouse ٠
- Dimensional data ٠
- Star and snowflake schema ٠
- Snowflake design issues ٠
- Developing the LDM: Selecting and classifying data ٠

# Attributes and Keys

- Domains
- Derived and calculated attributes ٠
- Multi-valued attributes
- Primary and foreign keys
- Normalization

# Modeling for Decision Support

- OLTP
- DSS
- Dimensional data ٠
- Star schema ٠
- Snowflake schema ٠
- Developing the LDM

# Advanced Entity Relationship Modeling

- Dependent and associative entities
- Subtypes and supertypes ٠
- Null attributes and recursive relationships ٠
- Naming objects

Course # 06-2000

# Logical Data Modeling and Relational Database Design Overview

# 1 day

# Description

This is a high-level, practical overview of the concepts, notation, and techniques of logical data modeling and relational database design. The course also includes a discussion of scalability and how to achieve scalability in design.

# Audience

IT professionals interested in relational database development including project managers, team or group leaders, analysts, designers, programmers, and support personnel. This course may also be valuable for business users participating in the design process.

# Prerequisites

Familiarity with computing systems and programming concepts is strongly recommended. Prior knowledge of the relational database model is helpful.

# **Classroom Requirements**

Overhead projector, whiteboard, flipcharts.

# Topics

# Introduction

# **Modeling Concepts**

- Introduction to modeling
- Modeling definitions and considerations
- Levels of modeling
- Types of modeling
- Zachman framework
- Modeling the system
- identifying models

# Logical Data Modeling

- Introduction to logical data modeling
- Logical data modeling approaches
- Developing a data dictionary
- Entity-Relationship modeling
- E-R modeling concepts
- Developing E-R models

# Format

- Presentation
- Written Exercises

# Objectives

After completing this course, participants should be able to:

- Understand modeling techniques used in relational database design
- Describe the differences between conceptual, logical, and physical
- models and their place in the phases of relational database design
  Describe logical data modeling techniques
- Describe logical uala modeling techniques
   Create interpret and apply aptity relationship a
- Create, interpret and apply entity-relationship diagrams
- Describe relational database design concepts
- Recognize the implications of scalability for modeling during the design phase

# **Relational Database Design**

Introduction to relational database design

# Scalability

- Scalability concepts
- Conclusion

# **Relational Database Design**

#### 2 days

#### Description

This course explains how to design a relational database using database design models and principles. Participants will learn how to refine an initial database design through various concepts. The course also discusses ways to reduce data redundancy and utilize logical design methods to "tune up" designs.

# Audience

Systems analysts/designers and application developers.

#### Prerequisites

Familiarity with the concepts and practices of logical data modeling as taught in Arkoa's *Logical Data Modeling* course is required. A basic understanding of SQL is desirable but not necessary.

Course # 06-2100

Databases

# Format

- Presentation
- Written Exercises

# Objectives

After completing this course, participants should be able to:

- Describe the concepts of good relational database design and its benefits
- Recognize the roles involved in database design
- Refine initial database table design
- Perform advanced relational database design
- Develop a first-cut data warehouse design

#### Topics

# Introduction

- Database design process
- Logical data modeling vs. database design
- Three data model levels
- Roles and responsibilities

# **Relational Databases**

- Entity-relationship LDM concepts
- E-R models
- Primary and foreign keys
- Concurrency control
- Security
- Optimizer
- Physical storage of tables

# Logical Database Design

- Normalization
- ♦ 1<sup>st</sup> through 5<sup>th</sup> normal form
- Domains or data types
- Detailed table design
- When to denormalize?
- Contrived columns or artificial keys
- Redundant or derived tables
- Data partitioning
- Mapping supertype/subtype entities to tables

# Transactions vs. Decisions

- Data warehouse design considerations
- Dimensional data
- Physical data warehouse design

# Physical Database Design

- Indexes
- Clustered vs. unclustered indexes
- Index storage structures: B-tree and hash
- Database sizing
- Database-level options for security design
- Integrity

# Introduction to MS SQL Server 7.0

# 5 days

**Databases** 

# Description

This course introduces the major features and functions of MS SQL Server 7.0 with a focus on using SQL for the development of single- and multitable queries and report generation. An overview of Transact-SQL and stored procedures is also provided.

#### Audience

Anyone who will be using SQL Server data sources.

#### Prerequisites

A good understanding of relational database concepts is helpful but not required.

#### Hardware and Software

A PC network with Microsoft SQL Server 7.0 available. MS SQL Server client tools installed. Windows 95/98/ME/2000 or NT clients are appropriate.

# Format

- Presentation
- 🖪 Hands-on
- Demonstration

# Objectives

After completing this course, participants should be able to:

- Describe the history and use of SQL
- Describe the SQL Server environment
- Use SQL Server tools and utilities
- Query selected data
- Format result sets
- Evaluate queried data using functions and expressions
- Join data from multiple tables
- Develop queries to answer business questions
- Use Transact-SQL in queries
- Execute stored procedures

# Topics

# Microsoft SQL Server

- Introduction to SQL Server architecture
- Back-end development tools
- SQL Server Enterprise Manager

# **Relational Databases and SQL**

- Relational database concepts
- Normalization
- SQL review

# **SQL Server Datatypes**

- Datatypes and Nulls
- SQL Server data type conversions
- Binary large object data types

# **Basic Data Structures**

- Databases
- Database information
- Tables and Views

# **Basic Integrity Structures**

- Column-based integrity
- User-defined data types
- Rules
- Default values
- Indexes and constraints

# SQL Extensions in SQL Server

- Transact SQL
- SQL Server functions
- The Full Text Index

# **Database Access and Preservation**

# **SQL Server Security**

- Database roles
- Backup and restore

#### **Transactions and Locking**

- Transactions
- ♦ SQL Server locking
- Lock granularity

# Microsoft SQL Server 7.0 Programming and Optimization

#### 3 days

#### Description

This 3-day class teaches practical techniques for developing and troubleshooting high performance SQL queries and optimizing the performance of Microsoft SQL Server 7.0.

#### Audience

Application designers, developers and programmers and database administrators who will be developing solutions with Microsoft SQL Server 7.0.

#### Prerequisites

Participants must have prior knowledge of SQL. Database administration experience is helpful but not required.

#### Hardware and Software

Microsoft SQL Server 7.0 properly installed on Windows NT/2000. Windows 95/98/NT/2000 clients. Administrator privileges must be available for student exercises.

#### Format

- Presentation
- 🖪 Hands-on

# Topics

# **Session Configuration**

- Configure SQL Server defaults
- Set database options
- Configure ANSI SQL-92 compliance

#### Programming with Transact-SQL

- Basic programming constructs
- Batches
- Conditional statements
- Transactions

#### Stored Procedures

- Write and debug stored procedures
- Stored procedure return codes
- Execute remote procedures

#### Cursors

- Cursor types
- Declaring cursors
- Opening and closing cursors

# Triggers

- Trigger uses
- Trigger creation
- INSERT and DELETE in triggers
- Savepoints

#### **Declarative Referential Integrity**

- Enforce primary and foreign key integrity with triggers
- · Constraints and indexes

#### Improving SQL Query Performance

- Views and performance
- Data type conversions
- Join processing
- Subqueries
- Union queries

Course # 06-0350

# Objectives

After completing this course, participants should be able to:

- Configure SQL Server for performance
- Describe Transact-SQL
- Write stored procedures
- Develop cursors and optimize their performance
- Develop efficient stored procedures
- Develop triggers
- Enforce referential integrity
- Describe how SQL Server 7.0 processes queries
- Troubleshoot queries to improve performance
- Improve the performance of SQL statement processing
- Optimize SQL Server 7.0 tables and indexes
- Optimize join processing
- Write more efficient subqueries

# MS SQL Server 7.0 Database Administration

# 5 days

This course introduces the basics of installing, monitoring, and maintaining Microsoft SQL Server and prepares participants to administer a Microsoft SQL Server database.

# Audience

Anyone who needs to administer a Microsoft SQL Server database.

#### Prerequisites

Participants must have prior knowledge of SQL and of relational database theory.

#### Hardware and Software

A PC network with Microsoft SQL Server 7.0 available. MS SQL Server client tools installed. Windows 95/98/ME/2000 or NT clients are appropriate. Administrator privileges must be available for student exercises.

# Format

- Presentation
- 😐 Hands-on

# Objectives

After completing this course, participants should be able to:

- Describe database administration procedures
- Identify the features of Microsoft SQL Server
- Install and monitor Microsoft SQL Server

# Topics

# Setting Up SQL Server

 Overview of SQL Server Administration Application Architecture SQL Server Architecture Benefits of the SQL Server Architecture SQL Server Features What Comes with SQL Server Overview of SQL Server Databases

The Role of the Database Administrator Installation, Configuration, & Upgrades

- Planning an Installation Preparing the System for Installation SQL Server Installation Performing an Unattended Installation Testing the Installation Upgrading From Previous Versions Performing the Upgrade Starting and Stopping the SQL Server
- Creating Databases

   Database Structures
   Database Files
   Planning the Database
   Creating the Database
   Viewing Database Information
   Creating Tables and Adding Data
   Altering and Databases
   Shrinking a Database
- Transferring Data Data Transfer Methods Using INSERT for Data Transfer Bulk Insert Using Bulk Copy Process (bcp) Data Transformation Service (DTS)

# **Basic Administration Tasks**

- Ensuring System Security What is Security NT Authentication SQL Server Authentication Changing the Login Password Roles Assigning Server Roles Database Roles User Defined Database Roles Application Roles Accessing Databases Special Users Permissions Granting Permissions Denying User or Group Access Viewing Permissions Permission Chains Backing Up and Restoring Databases Backup Overview Adding a Backup Device Backing Up the Database Restoring the Database Restore Options Monitoring the Server Tools Update Statistics Errorlog Troubleshooting Potential Problems DBCC
  - Deadlocking

# Replication

- Introduction to Replication
   *What is Replication Types of Replication Planning for Replication Setting Up Replication Configuring a Server as a Publisher Declaring Publications for Snapshot or Transaction Replication Defining Articles Creating a Snapshot Agent Configuring Subscribers Adding Pull Subscriptions* 
   Advanced Replication
   *Quarter Subscriptions Advanced Replication*
  - Configure Publishing and Distribution Wizard Creating a Publication Adding Subscriptions Managing Replication

# **SQL Server Maintenance**

- Alerts, Operators, and Jobs
   Using the SQL Server Agent
   Creating Alerts
   Creating Operators
   Setting Up Jobs
   Scheduling Jobs
   Using the Error Log
   The Application Log
- Preventative Maintenance Planning for Disaster Server Maintenance Database Maintenance Table Maintenance Index Tuning Wizard Checklists

# Introduction to SQL and Transact-SQL

# 1 day

**Databases** 

This course provides an introduction to SQL and Sybase's Transact-SQL. It is designed for all new users of SQL as well as those not familiar with Sybase's implementation of it.

# Audience

Programmers, database administrators, and database designers. It is also useful for end users who need to access data using SQL.

# Prerequisites

Familiarity with the fundamentals of relational databases is helpful.

# Hardware and Software

A SQL-compliant database with enough logins for the instructor and the students.

# Topics

# Introduction to Relational Database Systems

- Why databases?
- Advantages of database approach
- Relational model
- Database tools
- Client/server model

# Fundamental Concepts and Terminology

- Entities, attributes, and relationships
- Logical and physical data models
- Data model diagrams
- The relational model
- Data retrieval

# SQL Server Fundamentals

- Sybase SQL Server
- Client/server communication
- Open server
- Database security and integrity
- System tables and stored procedures
- Sessions

# Interactive SQL

- What is a batch?
- Getting started with ISQL
- Using Transact-SQL Script Files

# Structured Query Language (SQL)

Sybase Adaptive Server database

aged by a relational database

After completing this course, participants should be able to:

Write Transact-SQL statements to add, amend, and delete data in a

· Write SQL queries to retrieve, sort, summarize, and change data man-

History

Format

C Presentation

**Objectives** 

Written Exercises

- Usage and limitations
- ♦ Using SQL
- Identifiers
- Databases, users, and tables
- The language
- Data Definition Language (DDL)
- Data Manipulation Language (DML)
- ♦ Joins
- Aggregate functions
- Subqueries
- Analyzing data by grouping rows

#### 4 days

#### Description

This fast-track, hands-on course teaches students how to use Sybase Transact SQL Server for application development. This course is useful for programmers using 4GL tools, such as PowerBuilder or Visual Basic as well as those using high-level languages, such as C or C++.

#### Audience

Programmers and database administrators developing or managing applications in a Sybase environment.

#### Prerequisites

Attendance at Arkoa's *Introduction to SQL and Transact SQL* or a similar course, or practical experience with major features of ANSI SQL89 or SQL92 is required.

# Hardware and Software

Sybase Adaptive Server installed on Unix or Windows NT. The Sybsyntax database should be installed for on-line help with SQL. It should be possible to load a 3.5" DOS-formatted solution diskette for use in the course.

#### Topics

# Fundamental Concepts and Terminology

# **SQL Server Fundamentals**

- Sybase approach to client/server
- Sybase tables and stored procedures

#### Interactive SQL

Using Sybase ISQL

#### **Transact SQL Fundamentals**

• Defining, retrieving, and modifying data

#### **Defining Data**

- Data types and user data types
- Creating tables and indexes
- Constraints and rules
- Views
- Defaults

# • Security features

# **Manipulating Data**

- Predicates
- Subqueries
- Analyzing data by grouping rows
- Temporary tables
- String, date, mathematical, system, and string functions

# Format

- Presentation
- 🖪 Hands-on

# Objectives

After completing this course, participants should be able to:

- Use key features of Sybase Adaptive Server's Transact SQL
- Create, maintain, and manipulate database tables
- Optimize application performance through the use of procedural extensions to SQL, stored procedures, and triggers
- Enhance Adaptive Server performance

# **Transactions and Locking**

- Database recovery
- Deadlock and optimistic locking

#### **Procedural Extensions**

- Writing procedural code with SQL
- Using cursors

# Stored Procedures

- Creating and managing stored procedures
- Security, integrity, and performance

# Triggers

- Event-driven processing
- Creating, viewing, and executing triggers

# Performance Issues

- Factors, approach, and fundamentals
- Table, index, and query design
- Concurrency control

# Sybase Adaptive Server Administration

# 4 days

# Description

This course provides an opportunity to learn and practice the major tasks of Adaptive Server and database administration. It explains how to install, monitor and maintain Sybase Adaptive Server. It covers resource management, monitoring, troubleshooting, and maintaining a secure and reliable Adaptive Server environment.

# Audience

SQL Server administrators and database administrators.

# Prerequisites

Attendance at Arkoa's *Developing with Sybase Transact SQL Server* or a similar course, or practical experience with Sybase.

#### Hardware and Software

Sybase SQL Transact Server installed on Unix or Windows NT. The Sybsyntax database should be installed for online help with SQL. It should be possible to load a 3.5" DOS-formatted diskette for use in course exercises.

# Topics

# Introduction

# **Server Definition**

- Installation steps
- Starting, running, and shutting down the server

# **Resource Definition**

• Disk initialization and disk mirroring

# Databases

• Creating databases and segments

# Security

- Special users
- sysusers and syslogins
- Security, permissions, and ownership
- Role definitions and functions

# **Database Logging and Recovery**

- Transaction log
- Recovery options

# **Backup and Restore**

# Periodic Maintenance

• Server, database, and table maintenance

#### Utilities

• Script maintenance

# Format

- Presentation
- 😐 Hands-on

# Objectives

After completing this course, participants should be able to:

- Allocate resources
- Create databases and database segments
- Control access by applying security features
- Maintain server, database, and tables
- Backup and restore databases and logs
- Troubleshoot problems

# **Configuration and Tuning**

- Memory utilization
- Configuration variables and options

#### **Remote Server Management**

- Remote access and logins
- **Basic Application Performance Tuning**
- Sybase Optimizer

# Troubleshooting

• Common problems and solutions

#### **Development Standards**

Audit Server

# Using Oracle7 SQL\*Plus

#### 3 days

#### Description

This intensive course addresses the use of SQL\*Plus, the Oracle tool that provides an interactive implementation of SQL. The course teaches attendees how to connect to an Oracle7 database using SQL\*Plus and to perform data manipulation, reporting and administrative tasks. Attendees will also learn SQL, the standard language used for accessing relational databases.

#### Audience

Analysts, designers and programmers who are actively involved in or preparing to embark on the development of systems using the Oracle database. This course is also appropriate for users who need to gain an understanding of Oracle SQL in order to produce reports based on data held in an Oracle database.

#### Prerequisites

Participants are not required to have had experience with SQL or any other database management systems.

#### Hardware and Software

Oracle7 properly installed on any platform, with sufficient logins for the number of students and instructor.

# Topics

#### Introduction

- Databases
- SQL
- 4th generation software development
- The Oracle product set

# The Oracle Implementation of SQL and SQL\*Plus

- SQL basics
- Combining data from more than one table
- Grouping and aggregates
- Naming and datatypes
- Tables and indexes
- Constraints
- Data modification
- Multi-user considerations and views
- SQL functions
- Dates
- Reporting with SQL\*Plus
- SQL\*Plus files
- Users and privileges

# Tuning SQL

- The optimizer
- Optimizing alternatives
- Manually controlling optimization
- Optimization hints
- Shared SQL clauses
- Tuning FROM clauses
- Combining unrelated queries
- Indexes

Course # 06-0700

- Presentation
- Hands-on

Format

# Objectives

- After completing this course, participants should be able to:
- Apply relational database concepts
- Describe data grouping, aggregate functions, and SELECT statements
- Create new tables and indexes
- Insert, update, and delete data
- Utilize SQL functions and data handling
- Produce formatted reports
- Use SQL command files

# Using Oracle7 PL/SQL

# 2 days

This course provides participants with the skills necessary to make full use of the procedural extensions to SQL that are supplied with Oracle7. It is a practical hands-on course with more than half of the class time spent on exercises.

# Audience

Application designers, developers and programmers who will be moving to an Oracle environment.

# Prerequisites

Familiarity with Oracle's implementation of SQL to the level provided by Arkoa's Using Oracle7 SQL\*Plus course is required.

# Hardware and Software

Oracle7 properly installed on any platform, with sufficient logins for the number of students and instructor.

# Topics

# Introduction

- The purpose of PL/SQL
- Block structures and flow control
- ♦ IF and LOOP statements
- ♦ General comments on PL/SQL
- Advantages of using PL/SQL

# Variable and Constants

- Declaring variables
- Declaring constants
- Using variables
- Variable scope

# Cursors and Multi-Row SELECT Statements

- Declaring and using cursors
- Implicit cursors

# **Composite Datatypes**

- Tables
- Records

# **Exception Handling**

- Predefined exceptions
- User defined exceptions
- ♦ SQLCODE and SQLERRM

# Stored Procedures and Functions

After completing this course, participants should be able to:

Describe the rationale behind PL/SQL

Create stored procedures and functions

◆ Perform PL/SQL programming

Define composite datatypes

Use exception handling

Implement packages

Attach triggers to tables

- Stored procedures and functions
- Creating SQL accessible functions
- Testing procedures

#### Packages

Format

C Presentation

🖪 Hands-on

**Objectives** 

- The Customer Maintenance Package
- Using packages
- Dependencies
- Wrapped PL/SQL
- Supplied packages

# Triggers

- An UPDATE trigger
- Triggering statement
- Trigger restriction
- Trigger action
- Creating, enabling, and disabling triggers
- Compiling triggers

# **Oracle7 Database Administration**

#### 3 days

#### Description

This course is designed to teach participants how to create, maintain, and administer an Oracle7 database. Administering users, performing backups, recovering a database, and using the Data Dictionary are discussed. The course makes extensive use of the DBA tool SQL\*DBA. This course does not cover application building with Oracle. The tools used to create applications are discussed only in the context of application tuning and database administration.

# Audience

System administrators who will be supporting large Oracle databases, and application developers who will be developing database applications using Oracle.

#### Prerequisites

Familiarity (to a limited extent) with SQL, SQL\*Forms and PL/SQL is required.

#### Hardware and Software

Dedicated Oracle7 server properly installed on any platform, with sufficient logins for the number of students and instructor. Administrator logins and privileges are also required.

#### Topics

#### What is a Database Administrator?

- Responsibilities of a DBA
- Data storage
- Integrity and consistency
- DBA accounts
- Performance monitoring

#### **Oracle Architecture**

- System processes
- Multiple Oracle instances
- User process structure
- The system and program global areas

#### Creating & Maintaining a Database

- The SQL\*DBA tool
- Database states
- Instance management and modification
- Recovery
- Database configuration
- Creating a database

#### The Data Dictionary

Data dictionary structure

# Format

- Presentation
- 🖪 Hands-on

# Objectives

After completing this course, participants should be able to:

- Describe the role of the database administrator (DBA)
- Diagram the Oracle database architecture
- Create and administer an Oracle database
- Perform user administration in an Oracle database environment
- Backup and recover a database
- Use the Data Dictionary
- Use the Oracle SQL\*DBA tool

#### File Structures

- Oracle database files
- Tablespaces and segments
- Data segments and row formats
- Oracle data formats
- Estimating storage requirements

#### Users and Security

- Enrolling users
- Privileges and roles
- Users and resources
- Information hiding
- Auditing

#### **Backup and Recovery**

- Database failure modes
- Redo logs, control files, and rollback segments
- Taking backups
- Recovering the database
- Using EXPORT and IMPORT
- Example scenarios and remedies

#### **Distributed Databases**

- Implementing a distributed database
- Distributed transaction processing

Course # 06-0900

# **Using Oracle Forms 4.5**

# 3 days

This course teaches Oracle Forms 4.5 from the Oracle Developer 2000 toolset. This course is designed to establish the critical programming skills required by all developers working with Oracle.

# Audience

Application designers, developers and programmers who will be using Oracle Forms 4.5.

# Prerequisites

A familiarity with Oracle's implementation of SQL and PL/SQL to the level provided by Arkoa's *Using Oracle7 SQL\*Plus* and *Using Oracle7 PL/SQL* courses is required.

# Hardware and Software

Oracle7 with Forms 4.5 properly installed on any platform, with sufficient logins for the number of students and instructor.

# Topics

# Introduction to Oracle Forms

- What is Oracle Forms?
- Oracle Forms concepts

# **Building a Default Form**

- Opening a new form
- The form's properties
- The New block
- Viewing, saving, and running the form
- Default validation

# **Building a More Complex Form**

- Creating the first block and adding the second block
- Testing the form
- Tailoring items

# Further Features

- Defining a list of values
- Formatting data
- Using classes

# Triggers

- Event and block-level triggers
- Trigger contents and scope

# **Push Buttons and Triggers**

- Creating a control block
- Creating a push button
- The WHEN-BUTTON-PRESSED trigger
- Alerts and timers

# Format

- Presentation
- 🗕 Hands-on

# Objectives

After completing this course, participants should be able to:

- Develop and implement Oracle Forms 4.5 applications
- Create default Forms applications that retrieve, display, change, and store data
- Build a Forms application by customizing a default form
- Use the Object Navigator
- Add complex triggers to forms
- Use menus and libraries
- Debug applications

# Windows, Canvasses, and Procedures

- Creating a dialog window
- Creating a canvas
- Creating functions and procedures

# Further Notes on PL/SQL and Forms

- PL/SQL restrictions in Oracle Forms
- Key triggers and built-ins

# Block Relations

• Creating and using a relation

# Error Handling

- Detecting and raising errors
- Multi-User Considerations
- COMMIT processing and triggers; locking

# Further Features of Oracle Forms

• Global and system variables, message severity levels

# Menu Modules

• Creating menu items and sub-menus

# Libraries and Library Modules

- Creating and using a library
- Editing a library function
- Copying and referencing objects

# **Debugging and Documenting**

- Portability, block mode, and validation units
- The command line interface

#### 2 days

#### Description

This course teaches Oracle Reports 2.5 from the Oracle Developer 2000 toolset. This course is designed to provide the critical programming skills required of all developers working with Oracle.

#### Audience

Application designers, developers and programmers who will be using Oracle Reports 2.5.

#### Prerequisites

A familiarity with Oracle's implementation of SQL and PL/SQL to the level provided by Arkoa's *Using Oracle7 SQL\*Plus* and *Using Oracle7 PL/SQL* courses is required.

#### Hardware and Software

Oracle7 with Reports 2.5 properly installed on any platform, with sufficient logins for the number of participants and instructor.

#### Topics

#### Introduction to Oracle Reports

- Running Oracle Reports
- Data model objects
- Types of reports

# **Creating a Simple Report**

- Opening a new report
- Adding a query
- The report layout
- The layout editor
- Saving the report
- Running a report

# Master-Detail Reports

- Adding a second query
- Linking the queries
- Properties of links
- Formatting the report
- Report frames
- Adding a page header
- Page breaking
- Adding a total
- Combining fields and text

#### Adding Graphics and Text

- The layout editor
- Page numbers
- Importing graphics
- Linking text files
- Setting report properties

# Format

- Presentation
- 🖪 Hands-on

# Objectives

After completing this course, participants should be able to:

- Use Oracle Reports 2.5 to create simple reports as well as customized reports
- Add multiple queries to a single report
- Create nested and matrix reports
- Add graphics and text to reports
- Integrate triggers with reports
- Matrix Reports
  Creating a matrix report
- Creating and linking cross-product groups
- Matrix report frames
- Adding a summary
- Running the report

# **Further Features of Oracle Reports**

- Using triggers
- Report triggers
- Object level triggers
- Creating a format trigger
- System parameter queries
- Creating a user parameter and validation trigger
- Creating a report trigger to supply a default query value
- Extending Reports with lexical parameters
- Customizing the runtime parameter form
- Extending Reports to support detail drill-down

Course # 06-0420

# **Oracle Developer 2000**

# 5 days

**Databases** 

This course develops skills in the major tools used for application development in the Oracle Developer 2000 toolset. The tools covered include Oracle Forms 4.5 and Oracle Reports 2.5. This course is designed to provide the critical programming skills required by all developers working with Oracle.

# Audience

Application designers, developers and programmers who will be using Oracle Forms 4.5 and Reports 2.5.

# Prerequisites

A familiarity with Oracle's implementation of SQL and PL/SQL to the level provided by Arkoa's *Using Oracle7 SQL\*Plus* and *Using Oracle7 PL/SQL* courses is required.

# Hardware and Software

Oracle7 with Forms 4.5 and Reports 2.5 properly installed on any platform, with sufficient logins for the number of students and instructor.

# Topics

# Introduction to Oracle Forms

Oracle Forms concepts

# **Building a Default Form**

• Opening, viewing, saving, running, and simple tailoring

- **Building a More Complex Form**
- Using blocks

# **Further Features**

- Defining a list of values
- Formatting data
- Using classes

# Triggers

• Event triggers

# **Push Buttons and Triggers**

• Alerts and timers

# Windows, Canvasses, and Procedures

- Creating a dialog window
- Creating a canvas

# Further Notes on PL/SQL and Forms

- ◆ PL/SQL restrictions in Oracle Forms
- Key triggers and built-ins

# **Block Relations**

• Creating and using a relation

# Format

- Presentation
- 🗕 Hands-on

# Objectives

After completing this course, participants should be able to:

- Develop and implement Oracle Forms 4.5 applications
- Create default Forms applications that retrieve, display, change, and store data
- Build a Forms application by customizing a default form
- Use the Object Navigator
- Use menus and libraries
- Debug applications

# Error Handling

- Detecting and raising errors
- ON-ERROR triggers
- Multi-User Considerations
- COMMIT processing and triggers; locking

# **Further Features of Oracle Forms**

- Global and system variables
- Message severity levels
- User exits

# Menu Modules

- Creating a new menu module and sub-menus
- Integrating the menu into the form

# Libraries and Library Modules

- Creating, using, and editing
- Copying and referencing objects

# **Debugging and Documentation**

• Portability, block mode, and validation units

# Appendix A: The Example Company and Applications

• Appendix B: SQL\*Forms Reference Information
# Using Oracle8 SQL\*Plus

#### 3 days

#### Description

This intensive course addresses the use of SQL\*Plus, the Oracle tool that provides an interactive implementation of SQL. The course teaches attendees how to connect to an Oracle database using SQL\*Plus and to perform data manipulation, reporting, and administrative tasks. Attendees will also learn SQL, the standard language used for accessing relational databases.

#### Audience

Analysts, designers and programmers who are actively involved in or about to embark on the development of systems using an Oracle database. This course is also appropriate for users who need to gain an understanding of Oracle SQL in order to produce reports based on data held in an Oracle database.

#### Prerequisites

Participants are not required to have had experience using SQL or any other database management systems.

#### Hardware and Software

Oracle8 or 8i properly installed on any platform, with sufficient logins for the number of students and instructor.

#### Topics

#### Introduction

- Introduction to SQL The relational model and RDBMS Oracle DBMS architecture SQL\*Plus
- The Cars database

#### **Single Table Queries**

- Projecting data with SELECT The SELECT statement Developing and executing SQL statements Modifying results with expressions Incorporating functions in queries Working with NULL values Grouping related rows
- Restricting data with SELECT Comparing expressions Inclusion with IN Restricting result sets Sorting results

#### **Multiple Table Queries**

- Joining data
- Advanced joins and multi-table operations

#### Modifying Data

♦ INSERT / UPDATE / DELETE

#### Format

- C Presentation
- 🖪 Hands-on

## Objectives

- After completing this course, participants should be able to:
- Apply relational database concepts
- Describe data grouping, aggregate functions, and SELECT statements
- Create new tables and indexes
- Insert, update, and delete data
- Utilize SQL functions and data handling
- Produce formatted reports
- Use SQL command files
- Explain Oracle8's object relational features

#### **Physical Design and Views**

- Reconstructing business objects
  - What is a view? Recreating subtypes Recreating supertypes Row-level security View restrictions

#### Indexes

 Index design and maintenance Creating an index Index design

#### SQL\*Plus Report Writing

 Creating and formatting reports *Reporting with SQL\*Plus BREAK processing*

#### Introduction to Oracle8 Objects

Introduction to objects

#### **Tuning Oracle**

- Optimizing queries and indexes *The Oracle Optimizer SQL\*Plus AUTOTRACE Using TKPROF with TRACE*
- Index performance

Course # 06-0710

# Using Oracle8 PL/SQL

# 2 days

This course provides participants with the skills necessary to make full use of the procedural extensions to SQL that are supplied with Oracle. It is a practical hands-on course with many exercises.

#### Audience

Application designers, developers and programmers who will be moving to an Oracle environment.

#### Prerequisites

Familiarity with Oracle's implementation of SQL to the level provided by Arkoa's Using Oracle8 SQL\*Plus course is required.

#### Hardware and Software

Oracle8 or 8i properly installed on any platform, with sufficient logins for the number of students and instructor.

## Format

- Presentation
- 😐 Hands-on

#### Objectives

After completing this course, participants should be able to:

- Describe the rationale behind PL/SQL
- Perform PL/SQL programming
- Define composite datatypes
- Use exception handling
- Create stored procedures and functions
- Implement packages
- Attach triggers to tables

#### Topics

#### Introduction

Introduction to PL/SQL
 The purpose of PL/SQL
 Advantages of using PL/SQL
 Oracle8 object relational features

#### **Fundamentals**

- Block structures and flow control IF, Loop, and GOTO statements
- Variables and constants Declaring variables Declaring constants Using variables Variable scope

#### **Composite Data Types**

- Standard PL/SQL methods: RECORD and TABLE
- Object-relational methods: reference types and collection types

#### **Cursors and Exception Handling**

- Cursors
  - Declaring and using cursors Implicit cursors Cursor variables
- Exception handling *Predefined exceptions* User defined exceptions
- ♦ SQLCODE and SQLERRM

#### **Stored Procedures and Functions**

- Stored procedures
- Stored functions
- Creating SQL accessible functions
- Using functions and procedures
- External procedures

#### Triggers

- Components of a trigger The trigger statement The restriction The trigger action
- Using triggers
  - Creating, enabling, and disabling triggers Compiling triggers

#### Packages

- Using packages EXECUTE permission Dependencies Wrapped PL/SQL
- Supplied packages
- Oracle packages DBMS\_PIPE package DBMS\_OUTPUT package UTL\_FILE package
- ♦ DBMS\_JOB package

# **Oracle8 Database Administration**

#### 3 days

#### Description

This course teaches participants how to create, maintain, and administer an Oracle8 or 8i database. Administering users and storage, performing backups, and recovering a database are covered. Security and database optimization are also discussed.

The course makes extensive use of the DBA tool SQL\*DBA and the Server Manager. This course does not cover application building with Oracle.

#### Audience

System administrators who will be supporting Oracle databases, and application developers who will be developing database applications using Oracle.

#### Prerequisites

Some familiarity with SQL and PL/SQL is required.

#### Hardware and Software

Dedicated Oracle8 or 8i server properly installed on any platform, with sufficient logins for the number of students and instructor. DBA and Administrator log-ins and privileges are also required.

#### Topics

#### **Basic Concepts**

- ♦ Oracle architecture
- Logical structure and storage structures
- Database administrator tools

#### Creating the Database

- Initial planning
- Steps to create an Oracle database ٠
- After database creation ٠

#### Database Control

- Starting and shutting down the database
- Altering database availability

#### Managing Database Storage

- Setting storage parameters
- DDL, partitions, and optimization
- Managing tablespaces and datafiles
- Renaming and relocating tablespaces

#### **Oracle8 Objects**

- Object type definition
- Object tables, views, and comparison
- LOBs, BLOBs, and CLOBs

#### Format

- Presentation
- Hands-on

#### Objectives

After completing this course, participants should be able to:

- Describe the Oracle database architecture
- Describe the role of the database administrator (DBA)
- Create and administer an Oracle database ٠
- Manage Oracle8 storage structures ٠
- Describe Oracle8 objects ٠
- ٠ Perform user administration in an Oracle database environment
- Backup and recover a database ٠
- Describe Oracle8's security mechanisms ٠
- Use SQL\*DBA and Server manager ٠
- Optimize an Oracle8 database

#### Managing Rollback Segments

- Assignments to rollback segments
- Types of rollback segments
- Acquiring of rollback segments
- Changing storage parameters

#### Redo Logs and Control Files

- Managing and archiving redo logs
- Creating and dropping control files

#### **Database Security**

- Oracle licensing and security guidelines
- Creating, altering, and dropping users and roles
- Profile management
- Granting object privileges

#### Backup and Recovery

- Performing full and partial backups
- Recovery mechanisms and datafile restoration

#### Export and Import

- EXP and IMP operation notes
- System backup management

#### **Database Optimization**

- Tuning memory, hit ratio, and tuning I/O
- Managing data storage

#### Description

This intensive course provides a solid introduction to the Java programming language and development environment. In this fast-paced, handson course, programmers acquire the skills they need to write standalone, client/server, and intranet applications in Java.

#### Audience

Programmers who plan to develop Java applets and applications.

#### Prerequisites

Java is an advanced object-oriented programming language. Programming experience and familiarity with C are required. An understanding of C++ is desirable. Participants should understand object-oriented concepts and have used a WWW Browser, such as Netscape or Explorer.

#### Hardware and Software

PCs or workstations capable of running the Java Software Development Kit (SDK) version 1.2 or above. The systems should be able to load a 3.5" DOS-formatted solution disk.

#### Topics

#### Java Environment

#### Java Basics

- Java vs. C++
- Variables, data types, statements, flow control
- Simple output

#### **Classes in Java**

- Instantiating a class
- Member access, class definition, method overloading, constructors, encapsulation, and finalization

#### Arrays and Strings

#### Inheritance

 Protected access, Overriding methods, constructor chaining, abstract classes, interfaces

#### Programming with AWT-Applets

#### **Event Handling and Inner Classes**

- Events
- Listener interfaces
- Action events, adjustment events, item events
- Low-level events
- Inner classes revisited

#### Visual Dynamics with Applets

Images, threads, audio

#### **AWT Components**

 Labels, buttons, text components, lists, choice menus, checkboxes, scrollbars

#### Format

- Presentation
- 🖪 Hands-on

#### Objectives

After completing this course, participants should be able to:

- Describe the context for and uses of Java
- Describe the Java environment and steps needed to create a Java program
- Use Java to create applets
- Use Java to develop applications

#### AWT Layout Managers and Containers

- Layout and layout managers
- FlowLayout, GridLayout, GridBagLayout, BorderLayout
- ♦ Panel class
- Canvas class
- Frames, menus, dialog boxes

#### Exceptions

throw, catch, finally, throws

## I/O Streams

#### Collections, Wrapper Classes, and Cloneable Interface

- Cloning an object
- Vector class, Stack class, Hashtable class
- Object wrapper classes

#### **Multiple Threads**

- ♦ Class Thread
- Thread synchronization
- wait() and notifyAll()

#### **Appendices**

- ♦ HTML Guide
- ♦ Java sample application
- Java class hierarchy

# Advanced Java2 Programming

#### 5 days

#### Description

This course is intended for programmers who are familiar with the Java environment and syntax and are ready to learn more about developing complex applications. It also discusses some of the issues involved in the development and deployment of distributed applications.

#### Audience

Programmers who develop Java applications.

#### Prerequisites

Attendance at Arkoa's 5-day Java Programming course or equivalent training or experience.

#### Hardware and Software

PCs or workstations capable of running the Java Development Kit (JDK) version 1.2 or above, JDBC, Java RMI, and Java IDL. The systems should be able to load text files from a 3.5" DOS-formatted solution disk.

#### Topics

#### JavaBeans

- The BeanBox tool
- Events
- the core reflection API
- indexed and bound properties
- object serialization
- ♦ JAR files
- introspection

#### Multithreading

- Creating threads
- Inheriting from the thread class
- Thread scheduling
- Priority
- Synchronization
- Thread completion
- Stopping a thread
- Daemon threads
- The ThreadGroup class
- Thread states

#### **Networked Applications**

- TCP/IP overview
- ♦ UDP
- ♦ TCP
- Network Addressing and Port Numbers
- Class InetAddress
- Sockets
- The ServerSocket class
- Reading and Writing binary data
- Multithreaded servers
- The DatagramPacket class
- Datagram sockets
- The URL classes

#### Format

- Presentation
- 🖪 Hands-on

#### Objectives

- After completing this course, participants should be able to:
- Design and develop a Java application
- Implement ANSI SQL queries to a relational database
- Discuss the JDBC-API
- Program Java network connections and interfaces
- Recognize the use of alternative distributed computing solutions
- Define application-level security policies

#### JDBC

- ♦ JDBC versus ODBC
- JDBC and SQL
- ♦ JDBC architecture
- ♦ JDBC driver types
- ResultSets
- Using streams
- PreparedStatement
- CallableStatement

#### **Remote Method Invocation**

- RMI features
- Distributed vs nondistributed models
- Configuration scenarios
- RMI through firewalls
- Compiling and deploying files
- Starting the registry, server, and applet

#### Using Native Methods

- Overview of Java's native interfaces
- Writing the C function
- Java to C data conversions
- JNI interface function table
- Accessing object member variables
- Accessing arrays
- String operations
- Calling a Java method from C
- Method signatures

#### **Understanding Java Security**

- The mocha decompiler
- Public key encryption
- Digital signatures and certificates
- Java's security package
- Enhanced capabilities of trusted applets
- The Java Security Manager

# Introduction to Web Development with Java

#### 2 days

#### Description

This course provides technical managers and application developers with an introduction to Web site development with Java technologies. It surveys the different technologies involved in developing Web-based applications, with a focus on Java applets, Java servlets, JavaServer Pages (JSP), and Java Database Connectivity (JDBC). In addition to these technologies, the interaction of Java applets with JavaScript and the interaction of JSP with JavaBeans and Enterprise JavaBeans (EJB) are covered.

#### Audience

Java

Technical managers and application developers interested in developing Web-based applications using Java technologies, and/or interested in communicating with developers and architects of such applications.

#### Prerequisites

Familiarity with HTML, SQL, and at least one high-level programming language (such as Fortran, COBOL, Pascal, or C) is recommended.

#### Hardware and Software

PCs or workstations capable of running the Java Development Kit (JDK) and Java 2 Enterprise Edition (J2EE) developers kit. The systems should be able to load setup software from a CD-ROM disk.

The standard course setup requires a Windows 95/98/ME/NT/2000 or UNIX workstation. A Microsoft Access database is provided for JDBC exercises, for Windows systems.

#### Format

- Presentation
- 🖪 Hands-on

#### Objectives

After completing this course, participants should be able to:

- Define and describe terms and concepts used to communicate about Java technologies
- Communicate with Java developers and application architects about the design of Java applet, servlet, JSP, and JDBC code
- Describe the role of each Java technology in Web-based application architecture
- Interpret (read) existing Java applet, Java servlet, JSP and JDBC code, at a basic level
- Outline the basic structure of Java Web-based applications
- Identify which of these technologies to study at a deeper level, if necessary

#### Topics

#### Architectural Overview

- Web application architecture concepts and terminology
- Object-oriented concepts for Java
- Survey of Java and related technologies

#### **Getting Started with Java Applets**

- · Syntax and program structure overview
- Compilation and running programs
- Working with variables and literals
- Using expressions, statements, and control structures
- Working with classes
- Getting started with Abstract Windowing Toolkit (AWT)

#### Enhancing Web Pages with Java Applets

- Embedding Java applets in HTML
- Interacting with the user
- Handling events
- Using JavaScript with Java applets

#### Extending the Web Server with Java Servlets

- Using Java servlets (overview)
- Managing servlet life cycle
- Interacting with clients
- Saving client state

#### Server-Side Scripting with JavaServer Pages (JSP)

- Using JSP (overview)
- Using JavaBeans from JSP
- Working with server objects from JSP
- Using Enterprise JavaBeans from JSP

#### Connecting to Databases with JDBC

- Using JDBC (overview)
- Handling database communication
- Using database connection pooling
- Working with databases from JSP

66

# Java Foundation Classes (JFC) and Swing Development

#### 4 days

#### Description

This intensive course provides a solid introduction to using JFC to build portable GUI applications. Course attendees will use JFC and Swing components to build lightweight components and professional applications and applets.

#### Audience

Programmers who plan to develop Java applets and applications.

#### Prerequisites

Attendance at Arkoa's 5-day Java Programming course or equivalent training/experience.

#### Hardware and Software

PCs or workstations capable of running the Java Software Development Kit (SDK). The systems should be able to load a 3.5" DOS-formatted solution disk.

## Format

- Presentation
- 🖪 Hands-on

#### Objectives

After completing this course, participants should be able to:

- Design GUI architectures using MVC
- Utilize powerful Swing components in Java applications and applets
- Develop flexible GUIs through JFC's pluggable look and feel
- Create custom components (JavaBeans)
- Build multiple document applications
- Describe the assistive technologies available with Java and JFC
- Define techniques for international applications

#### Topics

#### JFC Swing: Fundamentals

- Pluggable Look and Feel (PL&F)
- Lightweight components
- Separable model architecture MVC
- ComponentUI in detail
- Jcomponent
- ContentPane, frames, and applets

#### JFC Swing: Layout Managers and Events

- Containers and Layout Managers
- New layout managers for swing
- Delegation Event Model
- Swing event types

#### JFC Swing: Buttons, Borders, Menus, and Toolbars

- Swing buttons
- Borders
- Menus and toolbars

#### JFC Swing: Trees and Tables

- Trees
- Tables

#### Panes, Text Handling, and Miscellaneous Components

- Panes, JrootPane
- JoptionPane
- Timers
- Progress bars
- Choosers

#### JFC Swing: Miscellaneous Features

- Clipboard
- Undo/Redo Framework
- Actions

#### JFC Accessibility

- Implementation
- Accessibility abstractions
- Accessible classes

#### JFC Java 2D

- Architecture
- JFC Drag and Drop
- Abstractions
- Transferable objects
- The drag and drop process

#### Java Performance Tuning

- The javac optimizer
- CLASSPATH issues
- JVM issues
- Optimizing for size
- Garbage collections
- Low-cost operations
- Strings and arrays
- Threads
- Streams
- Collections
- Miscellaneous techniques and idioms

Java

#### Description

This intensive, hands-on course provides an introduction to Web application development using Java Servlets and Java Server Pages (JSP). Students will learn how to create powerful server-side programs that benefit from the Java enterprise environment.

#### Audience

Java programmers or architects who plan to develop distributed, multi-tier Java Web applications. This course is particularly targeted to those developers writing Web server applications that utilize Java enterprise technologies.

#### Prerequisites

Attendance at Arkoa's 5-day *Java Programming* course or equivalent training/experience. Basic HTML knowledge equivalent to that provided by Arkoa's *HTML Programming* course is also recommended.

#### Hardware and Software

PCs or workstations capable of running the Java2 Software Development Kit (SDK) standard edition and enterprise edition and Tomcat 3.1.1 or above. The systems should be able to load setup software from a CD-ROM or floppy disk.

#### Format Presentation

😐 Hands-on

#### Objectives

After completing this course, participants should be able to:

- Design a multi-tier Java Web application
- Develop Java Servlet Web applications
- Design forms to post requests to Java based Web applications
- Implement Web applications with JSP technology

#### Topics

#### Multi-Tier Object-based Architectures

- Introduction to Objects in Distributed Systems Objects, Responsibilities, and Relationships Remote Object Interaction Proxy Pattern Proxies and Remote Objects Components
- Multi-Tier Architectures
   Distributed Software Systems
   Tier Components
   Web-based Tiers
   A Four-Tier Architecture
   Enterprise Development Architectures
   Microsoft DNA
   CORBA Architecture
   Sun J2EE Architecture
   J2EE Technologies

#### Introduction to J2EE

 J2EE Architecture and Java-based Application Servers The Enterprise Portal Myth Java 2 Enterprise Edition (J2EE) Java Application Servers Connecting to Legacy Data
 ♦ HTTP Technologies

The HTTP Protocol Programming Access to URLs HTTP HEAD, GET, and POST Requests HTTP Responses HTTP Status Codes

#### Introduction to Java Servlets

- Introduction to Servlets What is a Servlet? J2EE Web Containers and Web Applications Servlet Architecture
- Servlet Architecture javax.servlet.Servlet Interface Event Log ServletConfig Interface ServletRequest Class ServletResponse HttpServlet Class HttpServletRequest HttpServletResponse Sales Tax Calculator Example

#### Advanced Servlet Development Issues

 Considerations for Servlet Construction Error Handling Threading Life Cycle Issues Sessions and Servlet/Servlet Interactions
 Session Tracking
 URL Rewriting
 Cookies
 HttpSession Object
 Session Life cCycle Demonstration
 Shopping Cart Servlet Example
 ServletContext Interface
 NutShop: Setting the Inventory in the Context
 Additional ServletContext Methods
 RequestDispatcher
 Dynamic Content Generation

## Introduction to JavaServer Pages

 Introduction to JavaServer Pages Architecture Generating and Compiling the Servlet JSP Page Life Cycle JSP Tags Core JSP Tags: Declaration Core JSP Tags: Scriptlets Core JSP Tags: Expressions

#### Sophisticated Features

Implicit Objects Adding Comments to JSP Pages Using JavaBeans Setting and Getting Bean Properties The Include Directive The Forward Tag The JSP page Directive The taglib Directive The plugin Tag Forms Processing in JSP

## **JSP Custom Tags**

- Custom Tags Key Concepts Why Use Tag Extensions? JSP Anatomy: Tag Handlers Body Content
- More Custom Tags Scripting Variables

Body Tag Iteration Custom Tag Design Principles Custom Tag Usage

# JavaBeans Development

#### 3 days

#### Description

This intensive, hands-on course addresses the design and development of JavaBeans components. Through a series of lectures and lab exercises, course attendees will thoroughly explore building graphical JavaBeans software components and using JavaBeans in applications and applets.

#### Audience

Programmers who plan to develop JavaBean components or use Java-Bean components in applets or applications.

#### Prerequisites

Java

Attendance at Arkoa's 5-day Java Programming course or equivalent training/experience.

#### Hardware and Software

PCs or workstations capable of running the Java Software Development Kit (SDK) and Sun's Bean Development Kit (BDK). The systems should be able to load a 3.5" DOS-formatted solution disk.

#### Topics

#### JavaBeans Introduction

♦ JavaBeans: Basic Concepts

Java Classes and JavaBeans JavaBeans Advantages: Portability, Standard API, Simplicity Properties, Events, and Methods Introspection, Customization, and Persistence Run Time and Design Time Local Activation Beans and Threads Beans and Security Beans and Remote Access Visible and Invisible Beans Using JavaBeans: Applications, Applets, Web Page Content, and Components

• Getting Started with JavaBeans

The Beans Development Kit (BDK) Application Builders for Applet Construction Manual Applet Construction Simple Bean Walkthrough The BeanBox Uniform Windowing Data Transfer Review: JavaBeans Integrated Development Environment (IDE)

#### Events

JavaBeans Events

 Overview
 JavaBean Property Accessor Basics
 JavaBean Event Architecture
 Event State Objects
 AWT Events
 Custom Events.
 Generating Events
 Event Listener Interfaces
 Registering Event Listeners

Format Presentation

🗕 Hands-on

#### Objectives

After completing this course, participants should be able to:

- Design and implement JavaBeans
- Package JavaBeans components into JAR files and utilize those components in the BeanBox tool
- Use JavaBeans in an application or applet and implement event handlers
- Control the configuration of JavaBeans through property editors and customizers
- Implement a persistent JavaBean
- Describe the interoperability of JavaBeans with COM/ActiveX
  - Semantics of Event Delivery and Propagation Handling Events
- Event Adapters
   Event Adapter Overview
   Adapter Description
   Demultiplexing with Events
   Event Filtering
- Interlude: JavaBeans Additional Information Web Sites Other References Discussion Groups

#### Properties

- JavaBean Properties *Properties Overview Properties Accessor Methods Indexed Properties Exceptions in Accessor Methods Bound Properties Constrained Properties*
- Events with Bound and Constrained Properties Overview Listening to Bound and Constrained Events Responding to Bound and Constrained Events Races and Other Event Issues

#### Introspection, Customization, and Management

JavaBean Introspection Introspection Overview Reflection Design Pattern Overview Property Design Patterns Event Design Patterns Method Design Patterns The BeanInfo Interface and Classes Bean Analysis and the Inspector Class Inferred Names Security

 JavaBean Customization Customization Overview Customizer Interface Property Editor Interfaces The BeanInfo Interface

#### The BeanBox

Working with Beans in the BeanBox Editing Bean Properties Creating Property Relationships Connecting Beans with Event Handlers Creating an Applet The Tool Palette Bean Instantiation Testing Beans and Applets

#### JavaBean Persistence and Packaging

♦ Persistence

Persistence Goals What Should be Saved Serialization Interface Serializing Event Listeners Changes and Versioning Validating Persistent Objects Instantiating Persistent Objects

#### ♦ JAR File Usage

Packaging Requirements JAR File Overview BeanNames The JAR Program JAR File Representation

#### **Advanced Topics**

Distributed Bean Overview Distributed Bean Design Considerations Application-Component Communication with RMI Java IDL and CORBA JDBC as an API Java

#### Description

This intensive, hands-on course addresses the design and development of Enterprise JavaBeans (EJBs). Through a series of lectures and lab exercises, course attendees will thoroughly explore designing and building EJBs and using them in multi-tier, distributed systems.

#### Audience

Java

Java programmers or architects who plan to develop distributed, multi-tier Java applications. This course is particularly targeted to those developers writing server components for use in Web applications.

#### Prerequisites

Attendance at Arkoa's 5-day *Java Programming* course or equivalent training/experience. Attendance at Arkoa's 5-day *Advanced Java Programming* course is also recommended. Knowledge or experience in multi-tier application development to the level provided by Arkoa's *Multi-tier Client/Server Technical Overview* is helpful.

#### Hardware and Software

- Intel Pentium PCs with at least 128 MB of memory.
- Windows NT version 4.0 SP5 or Windows 2000 Professional
- Java 2 SDK, Standard Edition 1.2.2 or above
- Java 2 SDK, Enterprise Edition 1.2.1 or above

#### Topics

#### **Multi-Tier Object-based Architectures**

- Introduction to Objects in Distributed Systems
  - Overview

Objects, Responsibilities, and Relationships Responsibilities and Encapsulation Objects Interact with Each Other Remote Objects Typical in Enterprise Systems Remote Object Interaction Proxy Pattern Proxies and Remote Objects Components Benefits of Components Industry Components Language Independence Multi-Tier Architectures Overview Distributed Software Systems Tier Components Tier One Options Tier Two Options Tier Three Options Web-based Architecture Is Multi-Tier Web-based Tiers A Four-Tier Architecture Flexible Architecture

- Flexible Architecture Enterprise Development Architectures Microsoft Windows DNA CORBA Architecture
- Sun J2EE Architecture

  J2EE Technologies

## J2EE Platform

 Java 2 Platform, Enterprise Edition J2EE Specification J2EE Application Model J2EE Server Platform J2SE—The Foundation J2EE Presentation Components Servlets JavaServer Pages Clients J2EE Business Components EJB Relationships Containers ♦ J2EE System Services Overview Database and Transaction Services JTA J2EE Security JNDI JavaMail JMS RMI-IIOP Java IDL Deployment Example Application ♦ J2EE Platform Standards Overview J2EE Compatibility Test Suite Product Alternatives J2EE Reference Implementation

#### Format

- Presentation
- 😐 Hands-on
- Demonstration

#### Objectives

After completing this course, participants should be able to:

- Explain the advantages of Enterprise JavaBeans over other platforms for server-side component development
- Design Enterprise JavaBeans to implement the business tier of an ntiered application
- Develop scalable, robust server-side components using EJB
- Build secure, transactional component-based applications using EJB

#### EJB Architecture, Beginning EJBs

- EJB Architecture
   Overview
   Types of EJBs
   Inside the EJB Server
- Introduction to Implementing EJBs *EJB Interfaces and Classes RMI and EJB Interfaces Example Session Bean—Echo Deployment Descriptors*
- EJB Clients JNDI Overview EJB Clients and JNDI Echo Bean Client

#### Entity Beans—Container-Managed Persistence

- Container-Managed Persistence Entity Beans
  - Overview Persistent Store Alternatives Entity Bean Elements Possible Conflicts with Entity Beans Implementation Overview Entity Bean Life Cycle EntityContext
- Designing Container-Managed Entity Beans Overview Bean Attributes—Data Fields Implementing Primary Key
  - Defining the Home Interface Defining the Remote Interface Accessors/Modifiers Engineer Remote Interface Inquiry Remote Interface
- Implementing the Bean Class Implementing the Bean Class Using Data Objects Using Database Resources General Tips/Rules for Bean Implementation
- Deployment and Exception Handling Deployment J2EE RI—Deployment Entity Bean Clients Exception Handling

#### Entity Beans—Bean-Managed Persistence

- Designing Bean-Managed Persistence Bean-Managed Persistence Overview Entity Bean Life Cycle Implementation Overview
- Implementation Implementing the Bean Class
- Customer Example
   Customer Example Overview
   setEntityContext(), unsetEntityContext()
   ejbCreate()
   ejbFindByPrimaryKey()
   Collection ejbFindByEmailPattern()
   ejbStore()
   ejbLoad()
   ejbRemove()

Deployment
 Overview
 J2EE RI Deployment

#### **Stateless Session Beans**

- Session Beans Session Beans Overview Stateless v. Stateful Session Beans Choosing Session Bean Type
- Stateless Session Bean Architecture Stateless Session Bean Architecture Life Cycle Container Working Set Method Serialization Stateless Session Bean Elements
- Implementation
   Implementation Overview
   Interfaces and Classes
   SessionContext
   Transaction Management
   Environment Properties
   EJB References
- Designing Stateless Session Beans
   Overview
   CustomerAccountManagement Example
   Defining the Home Interface
   Defining the Remote Interface
   CustomerAccountManagement Remote Interface
- Implementing the Bean Class
   Implementing the Bean Class
   Attributes in Stateless Session Beans
   Implementing Remote Interface Methods
   Implementing the Home Interface Create Method
   Implementing SessionBean Callbacks
   Performance Tuning
- Deployment
   Deployment
   J2EE RI—Deployment
- Stateless Session Bean Clients Review

#### Transactions

- EJB Architecture and Distributed Transactions Transactions Overview ACID J2EE Support for Distributed Transactions Distributed Transactions Definitions Container's Role Single EJB Server Example Two EJB Servers and No Client Transaction Example Two EJB Servers and No Client Transaction Example Two EJB Servers and Client Transaction Example Voe EJB Servers and Client Transaction Example Client-Managed Transactions Overview Client-Managed Transaction Example Non-Transactional Beans
   Container-Managed Transactions Container-Managed Transactions
- Container-Managed Transactions Overview Container-Managed Transaction Context Bean Methods and Transaction Contexts EJBContext Methods Stateful Session Beans and SessionSynchronization

Transaction Attributes and Transaction Propagation Deployment Descriptor

- Bean-Managed Transactions Bean-Managed Transactions Overview Session Beans and Transactions Java Transaction API UserTransaction Interface Effect on Client Transactions Deployment Descriptor
- Transaction Isolation Issues and Exceptions Transaction Isolation Control Isolation Levels Setting Isolation Levels with JDBC Exceptions and Transactions

#### Stateful Session Beans

- Stateful Session Beans Overview and Deployment Overview Stateful Session Bean Architecture Life Cycle Activating/Passivating Stateful Session Beans Designing Stateful Session Beans CustomerInquiries Example State Machine for Bean Defining the Home Interface CustomerInquiries Remote Interface Designing the Bean Class
- Implementing and Deploying Overview Attributes in a Stateful Session Bean Implementing Remote Interface Methods Implementing the Home Interface Create Methods Deployment Stateful Session Bean Clients
- Transactions and Stateful Beans Overview SessionSynchronization Life Cycle with SessionSynchronization SessionSynchronization Callbacks

#### Security

- Security Overview Introduction Authentication Authorization Secure Communications
- J2EE Security Support J2EE Specification Declarative Security Programmatic Security Checks
- J2EE RI Access Control Demonstration Define Security Roles Assign Users/Groups to Roles Define Method Permissions J2EE Client Applications

#### JavaMail and JMS

♦ JavaMail Overview JavaMail Overview JavaMail Elements **Client Services** J2EE RI Support JavaMail API Major Components JavaMail Event Model ◆ JavaMail Programming Overview Implementation Outline ConfirmOrder EJB Mail Session Resource Factory JNDI Settings Using ConfirmOrder JMS Overview Messaging and JMS Typical Messaging Services Messaging Models JMS Concepts JMS Concepts Messages Destinations Connections and Connection Factories Sessions Message Producers Message Consumers Other Issues J2EE Requirements for JMS

#### What's Coming in EJB 2.0?

♦ EJB 2.0

EJB 2.0 Overview Integration with JMS Interoperability Between EJB Containers Updated Container-Managed Persistence

#### ◆ J2EE 1.3

J2EE 1.3 Overview JMS 1.0.2 J2EE Connector Technology 1.0 JAXP 1.1

# XML Programming with Java

#### 3 days

#### Description

This course focuses on the use of the JAXP 1.1 (Java API for XML Processing) for parsing and transforming XML documents with a Java application. This course covers the parsing and validation of XML using both the SAX (Simple API for XML) and DOM (Document Object Model) parsers and the integration of these parsers into Java-based applications. Additionally, the course covers the use of XSLT to transform XML documents into multiple formats.

#### Audience

The course is intended for Web and application developers who are interested in learning to use XML and Java together in order to leverage the platform independence and versatility of each.

#### Prerequisites

The course assumes a working knowledge of both XML and Java: for example, completion of Arkoa's *Introduction to XML* and *Java2 Programming*.

#### Hardware and Software

Each course participant needs a Windows 98, ME, NT, or 2000 PC, or a Unix Workstation with the following software:

- ♦ J2SE version 1.3 or above
- ♦ JAXP 1.1

#### Topics

#### XML and Java Overview

- XML Overview Introduction Components of XML DOM and SAX Comparison
- JAXP Overview Introduction Overview of JAXP 1.1 JAXP Parsers Package JAXP Transform Package

### SAX: Simple API for XML

 Introduction to SAX Introduction SAX Parser Products JAXP 1.1 Parser Support SAX Version 2 Introduction to Using SAX Creating Factory and Parser Parsing Introduction to Handlers Configuration Example Code

#### SAX Handlers

Introduction DefaultHandler Class ContentHandler Interface

#### Format

- Presentation
- 🖪 Hands-on
- Demonstration

#### Objectives

After completing this course, participants should be able to:

- Describe the differences between SAX and DOM parsers
- Describe the JAXP 1.1 API and its support for parsing and validating XML documents
- Use SAX and DOM parsers together with Java to manipulate and validate XML documents.
- Use XPath, XSL, and XSLT to control styles.
- Write XSL stylesheets and use XSLT together with Java to transform an XML document into multiple formats

Other Useful Interfaces Handling Errors and Warnings

#### **Document Object Model (DOM)**

- DOM API Overview
  - Introduction DOM Parsers DOM Nodes Overview DOM Tree Example Introduction to Using DOM Using JAXP to Support DOM
  - DOM Interfaces
  - DOM Error Handling
- XML Document Traversal and Modification Working with XML Documents Modifying a XML Document

#### XSL, XPath, and XSLT

- Introduction to XPath *Xpath Overview XPath Expressions Xpath Functions XPath Abbreviations*
- XSLT Introduction XSL Overview XSLT Overview Common XSLT Programming Constructs XSLT Processor

Java

 XSLT—Templates and Adding Elements and Attributes Working with Templates Controlling Output

## Transforming XML Documents Using JAXP

- Obtaining and Configuring a Transformer Introduction Using JAXP to Obtain a Transformer The Transformer Factory Handling Errors and Warnings URIResolver
- Transforming XML Introduction Using Transformer Using StreamSource or Result Using DOMSource or Result Using SAX Source or Result

# Internet Technical Overview

#### 1 day

#### Description

This technical overview provides a high-level introduction to the Internet. It describes Internet technologies and applications and discusses the benefits of the Internet as well as the issues that need to be addressed. It explains how to make productive use of the Internet in personal and/or business life. If the classroom is fitted with workstations and access to the Internet, this overview includes a demonstration and some hands-on exercises.

#### Audience

Technical managers and IT professionals who have little or no Internet experience.

#### Prerequisites

Familiarity with basic computer terminology is required.

#### **Classroom Requirements**

Overhead projector and flipcharts.

#### Topics

#### Introduction

- What is the Internet?
- The history of the Internet
- What is the World Wide Web?
- The history of the World Wide Web
- Internet connectivity
- Internet culture
- What is an intranet?

#### Internet Technologies

- Network technologies: transmission protocols and Internet access
- World Wide Web technologies: URLs and HTTP
- Web content creation technologies: HTML and DHTML
- Data access
- Security

## Format

- Presentation
- Demonstration

#### Objectives

After completing this course, participants should be able to:

- Use and understand Internet terminology
- · Explain how the Internet relates to the World Wide Web
- · Describe the benefits and issues involved in connecting to the Internet
- Describe the history of the Internet
- Describe Internet technologies and standards
- Describe popular Internet applications
- Use a popular Web browser

#### Internet Applications

- Electronic mail
- ftp and Telnet
- News
- Chat
- The World Wide Web: browsers, Web site management, and applications

#### Benefits and Issues

- Business benefits and electronic commerce
- Issues

#### 1 day

#### Description

This high-level overview explains how intranets can benefit work groups and companies. It describes the enabling technologies and popular products and explains the issues that need to be considered when planning, preparing for, or implementing an intranet.

#### Audience

IT managers and professionals interested in developing or assessing an intranet.

#### Prerequisites

Familiarity with basic computer terminology is required. High level understanding of Internet concepts is helpful.

#### **Classroom Requirements**

Overhead projector, flipcharts

#### Topics

Internet — Web

#### What is an Intranet?

- Web and Internet technology basics
- The Internet versus intranets
- Intranet content
- Intranet building blocks

#### The Benefits of an Intranet

- Intranet uses and benefits
- Intranets and groupware
- Intranet readiness factors

#### The Business Case for an Intranet

- Intranet start-up costs
- Cost saving areas
- Preparing an intranet cost justification

#### **Developing Intranet Applications**

- Overview of an intranet application
- Document publishing
- Intranet application development

#### Extranets

- Why are extranets needed?
- Issues raised by extranets
- Typical extranet applications

#### Intranet Components and Architectures

- Basic intranet components
- Web browser functionality
- Web server functionality
- Choosing Web browsers and servers
- Web architectures

#### Format

Presentation

#### Objectives

After completing this course, participants should be able to:

- Describe intranets, their history, and their relationship to the Internet
- · Identify the benefits and issues involved in implementing an intranet
- Describe intranet architectures and architectural components
- Explain intranet security threats, solutions, and issues
- Describe intranet applications and what an intranet can do
- Recognize the differences between groupware and an intranet

#### Planning for an Intranet

- Planning steps
- Strategy
- Analysis and design
- Implementation
- Creating a plan
- Marketing and evaluation

#### Building an Intranet

- Evaluate and select components
- Web server software and browsers
- Other services
- Installing components
- Laying out the Web site
- Web page design
- Building the Web site
- Implementing policies
- Setting up templates

#### Intranet Security

- Security requirements, policy, and mechanisms
- Data privacy mechanisms

#### Managing the Intranet

- Management and administration
- Webmaster and Web administrator roles

#### Advanced Intranet Applications

- Transaction processing
- Integration with groupware and existing systems
- Application design

# **Electronic Commerce: An Introduction**

#### 1 day

#### Description

The term "electronic commerce" means different things to different people. This course reviews these definitions and provides a summary of how business can be conducted electronically. This course also explores the business and technology considerations that are making the World Wide Web a popular vehicle for conducting business.

#### Audience

Anyone interested in electronic commerce, including technical managers and IT professionals.

#### Prerequisites

Familiarity with basic computer terminology is required. Familiarity with Internet/World Wide Web concepts is desirable.

# Topics

#### **Evolution of Electronic Commerce**

- Traditional versus electronic business
- History and growth of electronic commerce
- E-commerce models

#### Infrastructure for Electronic Commerce

- Networking technologies and protocols
- Internet architectures
- Internet service providers and services
- Enabling tools
- Selling and billing infrastructures
- Purchasing and order management infrastructures

#### **Electronic Commerce Applications**

- Business to business
- Business to consumer
- Industries and applications
- Case studies
- Legal issues

#### **Electronic Payment Systems**

- Types of business transactions
- · Forms of payment
- E-commerce standards
- EDI, EFT, and CALS
- SET
- Taxation

## Format

- Presentation
- 🖪 Hands-on

#### Objectives

After completing this course, participants should be able to:

- Identify the principal components of electronic commerce
- Compare traditional and electronic ways of doing business
- Recognize commonly used protocols and technologies utilized in an ecommerce site
- Understand differences between business-to-business and business-toconsumer e-commerce
- Differentiate between electronic payment protocols
- Identify large and small business e-commerce solutions
- Understand security risks and alternatives for securing e-commerce systems
- Plan the creation and management of an electronic business

#### **Commercial Merchant Solutions**

- Payment infrastructures (CyberCash, CyberSource, CyberTrust)
- Enterprise/large business solutions (IBM, Netscape, Microsoft, Sun)
- Small business solutions

## Security Considerations

- Risk areas
- Consumer privacy
- Securing payment processes
- Encryption
- Authentication
- ♦ Virtual Private Networks

#### **Electronic Commerce Business Opportunities**

- Creating awareness
- Marketing/selling products and services

#### **Establishing an Electronic Business**

- Scalable system design
- Integration with legacy systems
- Audit trails and reporting
- Managing the business

# Introduction to the Web and Web Development

#### 1 day

#### Description

This course explores the history, architecture, and tools of the Internet and the World Wide Web. The course provides a broad introduction to the tools and technologies behind the World Wide Web.

#### Audience

This course is designed for new Web developers or those interested in becoming Web developers. It is also beneficial for technical managers and IT professionals who have little or no Internet experience.

#### Prerequisites

Familiarity with basic computer terminology and experience using a Web browser is required.

#### **Classroom Requirements**

Overhead projector and flipcharts.

#### Format

- Presentation
- 🖪 Hands-on
- Demonstration

#### Objectives

After completing this course, participants should be able to:

 Describe the history and technical structure of the Internet and World Wide Web

Course # 99-9410

- Describe the architecture of the Internet and World Wide Web
- Explain fundamental standards and technologies related to the Internet and World Wide Web
- Define and explain intranets and extranets and compare them to the Internet and World Wide Web
- Define e-business and e-commerce and describe related technologies
- Describe Web development technologies and differentiate client-side and server-side development
- Describe the Web development process and the role of a Web developer

#### Topics

#### The Internet and the World Wide Web

- The Internet
- State of the Internet and World Wide Web

#### Internet/Web Protocols and Technologies

- Internet Protocols
- Accessing the Internet

#### Internet and Intranet Web Content

- Developing Web Page Content
- Intranet Content

#### **Dynamic Web Content**

- Client-Side Scripting
- Server-Side Scripting and Data Access

#### E-commerce

- ♦ Electronic Commerce
- Securing Electronic Commerce

#### **Corporate Web Site Development Overview**

- Planning and Design
- Production and Launch

# Introduction to Apache Web Server

#### 2 days

#### Description

This course covers the installation, configuration, and administration of the Apache Web server, the number one Web server used today. While Apache is available for numerous computing platforms, this course focuses on Unix in general and Linux in particular. Information on running Apache on Windows 95/NT is also provided.

#### Audience

New or inexperienced Webmasters or system administrators who are investigating or planning to set up an Internet or intranet site with Apache.

#### Prerequisites

Students should be familiar with the Unix operating system and have a basic knowledge of networking concepts.

#### Hardware and Software

Intel Pentium compatible PCs capable of running Linux and Perl, Apache Web server (which students will install), and a Web browser. If students wish to build Apache from source code, a C compiler such as GNU C++ is required.

## Format

- Presentation
- Hands-on

#### Objectives

After completing this course, participants should be able to:

- Install Apache Web server
- Configure Apache Web server
- Set up virtual hosts
- · Construct and use Perl scripts with Apache Web server
- Administer Apache Web server, including maintaining logs and implementing security

Course # 16-0100

#### Topics

#### Apache Architecture and Internet Refresher

- Architecture and features
- Internet refresher

#### Installation and Configuration

- Installation
  - Build and install Directory structure File and directory permissions Basic configuration Running Apache
- ♦ Configuration
  - Configuration files and file syntax Core module directives Adding modules Testing configuration changes

#### Virtual Hosts and Administration

- Virtual hosts
- SSI, URL rewriting, indexes, imagemaps
- Status and logging
  - Configuration information Access log Error log
- User tracking

#### **Common Gateway Interface**

- Perl
- Data types, variables, and operators
   Arrays and hashes, control statements
   Processes
   Matching
   User functions
   Input/output
   File and directory operations
   CGI and Perl
   Common Gateway Interface (CGI)
- Configuring Apache CGI examples, style, alternatives suEXEC

#### Authentication

- Basic authentication
- Secure socket layer

#### Firewalls, Proxies, Apache for Windows

- Firewalls and firewall architecture
- Apache proxy Proxy directives Caching directives Configuring browsers
- Apache for Windows

#### Description

This course provides an introduction to Web site design, covering both human factors and technical design issues. Guidelines for planning and designing a Web site are included.

#### Audience

Web designers and developers who need a practical introduction to Web site design.

#### Prerequisites

Participants should be familiar with using Web browsers and have experience in exploring the World Wide Web. HTML experience is helpful, but not required.

#### Hardware and Software Requirements

A PC or workstation capable of running IE 4.0+ or Netscape 4.0+.

# Format

- Presentation
- 🖪 Hands-on
- Demonstration
- Written Exercises

#### Objectives

After completing this course, participants should be able to:

- Define the structure of a Web site
- Apply human factors guidelines in site and page design
- Create a design plan for a Web site
- Apply layout techniques to page design
- Describe Accessibility problems in Web design
- Identify the challenges of targeting multiple browsers and platforms

#### Topics

#### Web Design Overview

- Introduction to Effective Web Design
- Effective Web Design Principles Visual Simplicity Consistency of Design Use of Common Visual Language Creating a Compelling Design Designing for Usability
- Designing for All Web Users Characteristics of User Needs Cross-Platform and Cross-Browser Design Load Time
- Planning an Effective Web Site

#### Page Design

- Effective Use of Color Basic Color Theory The RGB Color Model Color Depth Web-Safe Color Palette Color in HTML Designing with Color
- Page Layout with Tables
   Using a Table as a Layout Tool
   Minimizing Load Time
   Browser and Platform Compatibility with Tables
- Using Frames in Page Design Designing with Frames Browser Compatibility with Frames
- Page Layout with Style Sheets Cascading Style Sheets

- Effective Layout Design Applying Design Principles to Page Layout Page Size Design for "Above the Fold" Browser Offset Design for Printing
  - Page Design Dos and Don'ts

#### **Content Design**

- Text and Writing for the Web Typographical Design Text Formatting Using a Graphic Image as Text Text Layout
  - Writing for the Web
- Graphics

Effective Graphic Design Web Image File Formats Cross-Platform Image Design Editing Techniques Used with Graphics Optimizing Graphics Positioning Graphics on a Web Page Graphic Design Dos and Don'ts

 Multimedia and Interactivity in Web Design *Effective Use of Multimedia and Interactive Content Audio Files Video Files Using Animated GIFs on the Web Integrated Multimedia Content Using Forms*

#### Site Design

- Information Architecture
   Information Architecture Overview
   Content and Functional Requirements
   Content Organization
   Creating an Appropriate Structure for the Site
   Connections
   Information Architecture Dos and Don'ts
- Navigational Design

Navigational Design Objectives Navigation Layout Navigational Devices Navigation Dos and Don'ts

#### **Other Web Design Considerations**

- Accessible Web Design

   Introduction to Accessible Web Design
   Visual Impairments
   Hearing Impairments
   Motor Disabilities
   Accessibility Guidelines
   Accessibility Testing
- Web Design for a Global Audience Worldwide Distribution of Online Users Approaches to Global Web Design Language Usage Culture
- Intranet and E-Commerce Web Design Intranet Design E-Commerce Web Design Online Resources for Web Designers

# **HTML Programming**

#### 2 days

#### Description

This course provides developers with a practical, hands-on introduction to HTML document development. The course covers HTML page design and development using the most common and powerful HTML elements. The topics covered in this course are compatible with major Web browsers and HTML page development tools.

#### Audience

Intranet and Internet application designers and programmers who need to develop HTML pages. Web server application programmers who need an understanding of the interaction between client HTML pages and Web server applications.

#### Prerequisites

Familiarity with the World Wide Web and basic programming skills are recommended.

#### Hardware and Software

Windows 95/98/ME/NT/2000, version 4+ of preferred Web browser, and text editor.

#### Topics

#### Introduction

• Overview of a simple HTML page

#### HTML Page Structure

- Tags defined
- Head and body sections

#### **Basic Formatting with HTML**

- Formatting
- Layout of tags and lists

#### Links and URLs in HTML

- URL types and URL attributes
- Anchor tags

#### Images in HTML

- Image formats
- The <IMG> and <OBJECT> tags
- Creating an image map

#### Tables in HTML

- Basic tags for a table
- Advanced table layout
- HTML 4.0 tags

#### Frames in HTML

- Goal-oriented frame layout and tags
- Links between frames
- ♦ The <IFRAME> tag

#### Format

- Presentation
- 🗳 Hands-on

#### Objectives

After completing this course, participants should be able to:

- Develop HTML pages using static elements including form elements and tables
- Incorporate multimedia elements
- Add interactivity with scripting and components
- Explain how to add dynamic data to HTML pages with server processing

#### Forms in HTML

- <FORM>, <INPUT> and <SELECT> tags
- Form layout and processing

#### Server Processing and HTML

- Introduction to CGI
- The submit process
- Integrated server scripting

#### Multimedia in HTML

- Using color and background images
- Audio considerations for HTML
- Streaming video
- VRML

#### Scripting in HTML

- Javascript and JScript
- Embedded components in HTML

#### Style Sheets and DHTML

- Styles in HTML
- ♦ <DIV>, <LAYER>, and <SPAN> tags

# Internet – Web

#### Description

The HyperText Markup Language, or HTML, is the language of the World Wide Web. Proficiency with HTML is a requirement for any Web developer. This course provides students with a practical, hands-on introduction to HTML development using the most popular visual development tool: Macromedia's Dreamweaver.

#### Audience

Intranet and Internet application designers and programmers who need to develop HTML pages. Web server application programmers who need an understanding of the interaction between client HTML pages and Web server applications.

#### Prerequisites

Familiarity with the World Wide Web and basic programming skills are recommended.

#### Hardware and Software

PCs with 32 MB RAM and Windows 95, 98, 2000, ME or NT; Dreamweaver 3.0+ properly installed; text editor; any Web graphics program—Paint Shop Pro preferred.

#### Topics

#### **HTML Basics**

- Introduction to HTML
- ♦ HTML Page Structure
- Basic Text Formatting with HTML
- Links and URLs in HTML
- Web Site Structure

#### Introduction to Dreamweaver

- Dreamweaver Overview
- File and Site Management Using Dreamweaver
- · Formatting Text and Creating Links in Dreamweaver

#### Web Graphics

- Web Graphics Basics
- Optimizing Images
- Using Color and Images in HTML

#### **Tables and Frames**

- Tables in HTML
- Frames in HTML
- Usability and Web Design

#### Forms and Multimedia Elements

After completing this course, participants should be able to:

Use text, links, frames, tables, images, forms, and multimedia in Web

Explain fundamental graphic concepts and optimize graphics for the

• Use Dynamic HTML (DHTML) layers and positioning elements

Use Dreamweaver for Web page development

Forms in HTML

Format

Presentation

Demonstration

Develop Web pages using HTML

Use cascading style sheets

Hands-on

Objectives

pages

Web

- Using Forms in Dreamweaver
- Multimedia in HTML

#### DHTML and Cascading Style Sheets

- Introduction to DHTML
- Cascading Style Sheet Overview
- CSS Positioning and Layers

Internet – Web

#### Description

The HyperText Markup Language, or HTML, is the language of the World Wide Web. Proficiency with HTML is a requirement for any Web developer. This course provides students with a practical, hands-on introduction to HTML development using one of the most popular visual development tools: Allaire's HomeSite

#### Audience

Intranet and Internet application designers and programmers who need to develop HTML pages. Web server application programmers who need an understanding of the interaction between client HTML pages and Web server applications.

#### Prerequisites

Familiarity with the World Wide Web and basic programming skills are recommended.

#### Hardware and Software

PCs with 32 MB RAM and Windows 95, 98, 2000, ME or NT; Homesite 4.5+ properly installed; text editor; any Web graphics program—Paint Shop Pro preferred.

#### Topics

Internet — Web

#### **HTML Basics**

- Introduction to HTML
- HTML Page Structure
- Basic Text Formatting with HTML
- Links and URLs in HTML
- Web Site Structure

#### Introduction to Allaire HomeSite

- ♦ HomeSite Overview
- File and Site Management Using HomeSite
- Formatting Text and Creating Links in Home Site

#### Web Graphics

- Web Graphics Basics
- Optimizing Images
- Using Color and Images in HTML

#### **Tables and Frames**

- Tables in HTML
- Frames in HTML
- Usability and Web Design

#### Forms and Multimedia Elements

After completing this course, participants should be able to:

Use text, links, frames, tables, images, forms, and multimedia in Web

• Explain fundamental graphic concepts and optimize graphics for the

Use Dynamic HTML (DHTML) layers and positioning elements

Use Allaire HomeSite for Web page development

• Forms in HTML

Format

C Presentation

Demonstration

Develop Web pages using HTML

Use cascading style sheets

Hands-on

Objectives

pages

Web

- Using Forms in HomeSite
- Multimedia in HTML
- DHTML and Cascading Style Sheets
- Introduction to DHTML
- Cascading Style Sheet Overview
- CSS Positioning and Layers

# Advanced HTML and DHTML Programming

#### 3 days

#### Description

This course provides students with the skills and knowledge needed to create flexible, maintainable Web sites with advanced features. The course covers Cascading Style Sheets (CSS) and Positioning (CSS-P) and explains their use in creating more attractive Web sites that are easily maintained. The course also introduces XML and XHTML and its use in creating Web sites. Other advanced HTML elements are also discussed, including image maps, meta tags, components, and multimedia.

#### Audience

Anyone involved with the development of Web sites who wishes to learn CSS, CSS-P, or other advanced HTML features.

#### Prerequisites

Participants must have basic proficiency with HTML and JavaScript or have completed introductory HTML and JavaScript courses such as Arkoa's *HTML Programming* and *JavaScript for Non-Programmers* courses.

#### Hardware and Software

Windows 95/98/ME/NT/2000, version 4+ of preferred Web browser, and a text editor such as Notepad. Browser compatibility issues are discussed, so access to a second, different browser is preferable.

#### Topics

#### Meta Tags, Forms, and Image Maps

- Meta tags What are meta tags? Meta tag use
- Advanced form construction Creating usable forms Form element control Form processing
- Image maps

Creating image maps Implementing image maps

#### Integrating External Elements and Cross-Platform HTML

- Using components in Web pages Calling external components Using multimedia elements in HTML
- Cross-platform HTML issues *Platforms and browsers Design space Tables Frames*

#### **Basic Cascading Style Sheets**

- Cascading Style Sheet basics *Defining CSS CSS rules placement Adding CSS to an HTML document*
- Using Cascading Style Sheets
   CSS fonts
   CSS text control

#### Format

- Presentation
- 😐 Hands-on
- ♥ Interactive Activities

#### Objectives

After completing this course, participants should be able to:

- Use meta tags to describe and control Web pages
- Create forms adhering to usability guidelines
- Identify Cascading Style Sheet Rules
- Insert CSS rules into a document
- Create CSS tags
- Manipulate CSS properties
- Set CSS position attributes
  Create an image map
- Create an image map
- Identify cross-platform HTML rendering issues
- Insert a Multimedia element into a Web page
- Develop HTML code that will work in future browsers
  - Using style list properties Using color and background properties CSS margins and borders

#### Advanced Cascading Style Sheets and Positioning (CSS-P)

- Cascading Style Sheets Positioning (CSS-P) Using CSS-P for page layout Common positioning tasks
- Developing dynamic CSS
  - Changing attribute values via scripting Changing an element's class Providing alternate style sheets
- Cross-platform CSS issues Universally supported selectors Resolving cross-platform differences

#### HTML and the Future

- Creating well-formed documents !DOCTYPE definitions Deprecated tags and elements Obsolete tags and elements
- The Extensible Markup Language (XML) *XHTML XML*
  - XML and CSS
- XSL
- Privacy issues
   Privacy statements
   Platform for Privacy Preferences (P3P)
   Implementing P3P

Course # 02-7550

#### Description

JavaScript is perhaps the most popular language used for client-side scripting for Web development. This course provides students with a practical, hands-on introduction to JavaScript and its use in interactive Web pages.

#### Audience

Web developers and designers who plan on building interactive Web pages using JavaScript. This course assumes students have little prior programming experience.

#### Prerequisites

Completion of Arkoa's *Programming with HTML* course or equivalent knowledge is required. Understanding of basic programming concepts as provided by Arkoa's *Introduction to Programming* course is helpful.

#### Hardware and Software

Windows 95/98/ME/NT/2000, latest version of preferred Web browser, and text editor.

## Topics

Internet — Web

#### Introduction to JavaScript

- Introduction
- Embedding JavaScript in a Web Page Creating and Using JavaScript Files Writing to the Document
  - Functions
  - Debugging JavaScript
- Representing Data Data Types Escape Characters Data Variables Conversion Arrays

#### JavaScript Logic

- Expressions and Operators

   Types of Statements
   Comments
   Expressions
   Operators
   Functions -- Review
   Parameters vs. arguments Property
   Operator Precedence
- Control Statements

#### JavaScript and the Document Object Model

- JavaScript Object Model Using Objects
  - Browser Objects Document Object Document Object Hierarchy Custom Objects Events

#### Format

- Presentation
- 🖪 Hands-on
- Demonstration

#### Objectives

After completing this course, participants should be able to:

- Describe JavaScript syntax
- List and explain the HTML tags used with JavaScript
- Explain JavaScript's events and event handling process
- Write JavaScript programs to enable rollover functionality in Web pages
- Write JavaScript programs to validate form data
- Write JavaScript programs to control windows and frames
- Write JavaScript programs to handle common events
- Form Processing Coding Guidelines Form Objects
- Handling Events The Event Model Event Sources Event Types/Handlers
- Other Event Handlers

#### Manipulating the DOM with JavaScript

- Core Objects and Functions *Core Objects Top-Level Properties and Functions HTML Code Generation*
- Validating Form Input *Client-Side vs. Server-Side When to Validate Data? Creating a Regular Expression Common Pattern Matching Symbols Post-Pattern Modifiers Special Escape Pattern Characters Matching Regular Expressions*
- Document Control
   Image Object
   Image Selection
   Image Slide Show
   Image Roll
   Multiple Forms
   Replacing the Current Document
   Other Document Capabilities

#### Working with Windows, Frames, and Cookies

- Windows and Frames **Opening and Closing Windows** Referencing a Window Object Creating a Frame Updating a Frame Referencing a Frame Object window Object Properties window Object Methodwindow Object Events Handling Major Window Events
- Advanced Topics

JavaScript URLs The Status Bar What Is a Cookie? How Cookies Are Sent Using the cookie Property Using Cookies with JavaScript Custom Objects Defining a Custom Object Type Using Custom Objects

# JavaScript Programming

#### 2 days

#### Description

This course provides programmers with a practical, hands-on introduction to JavaScript and its use in interactive Web pages. The course explores JavaScript's built-in objects and capabilities as well as the creation of new objects.

#### Audience

Web developers and designers who plan on building interactive Web pages using JavaScript. This course assumes students have some prior programming experience.

#### Prerequisites

Completion of Arkoa's HTML Programming course or equivalent knowledge is required. Prior experience with a programming language, such as C, Java, or Visual Basic, is highly recommended.

#### Hardware and Software

Windows 95/98/ME/NT/2000, latest version of preferred Web browser, and text editor.

#### Format

- C Presentation
- Hands-on
- Demonstration

#### Objectives

After completing this course, participants should be able to:

- Describe JavaScript syntax
- List and explain the HTML tags used with JavaScript
- Explain JavaScript's events and event handling process
- Write JavaScript programs to enable rollover functionality in web pages
- Write JavaScript programs to validate form data
- Write JavaScript programs to control windows and frames
- Write JavaScript programs to handle common events

## Topics

#### Introduction

- What is JavaScript? Java versus JavaScript
- Objects in JavaScript

#### **Getting Started**

- HTML tags for JavaScript
- Basic command syntax
- Comments
- Displaying text and images
- Getting input from the user

#### **Basic Language Elements**

- Data types
- Variables
- Arrays
- Operators
- Flow control

#### **Creating Modular Code**

- Creating and using functions
- Creating and using objects

#### Events and Event Handlers

- Mouse events
- Simulating events

#### Forms

- Creating forms with HTML
- The JavaScript form object
- Form elements
- Working with multiple forms

#### **Built-in Objects**

- The date object
- The string object
- The math object

- Creating frames in HTML
- The document object
- The location and history objects

# **VBScript for the Web**

#### 1 day

#### Description

This course provides developers with a practical, hands-on introduction to Visual Basic Scripting and its use in interactive Web pages. The course explores VBScript's syntax and event-handling mechanisms, building interactive forms, and developing complex Web applications with multiple frames.

#### Audience

Intranet and Internet developers who plan on building interactive Web pages using VBScript.

#### Prerequisites

Completion of Arkoa's *Programming with HTML* course or equivalent knowledge is required. Prior experience with a structured programming language, such as C, Java, or Visual Basic, is recommended.

#### Hardware and Software

Windows 95/98/ME/NT/2000, latest version of MS Internet Explorer, and text editor.

## Format

- Presentation
- 🖪 Hands-on

#### Objectives

After completing this course, participants should be able to:

- Identify and use the basic elements of the VBScript language
- Create simple Web pages using VBScript
- Construct user interaction using VBScript and HTML forms

#### Topics

#### Introduction

#### VBScript Language

- Web scripting languages
- VBScript basics
- Invoking scripts
- Data types
- Declaring variables
- Arrays
- VBScript control structures
- Case selection and looping
- String functions
- Procedures and functions

#### **Browser Object Model**

- Window object and events
- Document object
- Link and anchor objects
- Form object
- Element object
- Collections
- History, navigator, and frame objects

#### Tools, ActiveX, and Active Server Pages

- FrontPage and scripting
- Debugging scripts
- ActiveX controls
- HTML layout controls
- Integrating client and server scripting

# ActiveX and COM Development

#### 5 days

#### Description

The course provides a solid introduction to Microsoft's Component Object Model (COM) and the ActiveX standard. Participants will use the Active Template Library (ATL) to develop ActiveX controls and COM components that are combined to create a multi-tiered, Web-based application. Course exercises are done using C++.

#### Audience

Experienced developers looking to create ActiveX components-based applications.

#### Prerequisites

A thorough understanding of C++ and experience with Windows programming. Familiarity with Visual C++ is useful.

#### Hardware and Software

Windows NT Server and Visual C++ 5.0 or higher. The systems should be able to load a 3.5'' solution disk.

#### Topics

Internet — Web

#### The Component Object Model

Unknown interface

#### Implementing COM

- Creating COM objects
- Class factories
- Server implementation issues

#### **Using Active Controls**

#### Implementing an Active Control

- ATL Wizard usage
- ATL Classes that support control creation

#### **Developing Custom Interfaces**

- Using IDL to describe/define an interface
- Implementing an interface

#### Implementing Business Rules

- Multi-threading support/issues—apartments
- Code reuse strategies
- Transaction processing support—MTS

#### **COM Objects Design Considerations**

- Designing a COM-based system
- Considerations for cost, speed, reliability, etc.

#### Automation and Type Information

- Dual interfaces
- Type libraries

#### Format

- Presentation
- 🖪 Hands-on
- Interactive Activities

#### Objectives

After completing this course, participants should be able to:

- Describe the COM architecture in detail
- Use the ATL Object Wizard to build ActiveX controls
- Develop a custom interface to a COM object
- Use COM development techniques to design and implement multitiered client/server applications

#### **Connectable Objects**

- ConnectionPointContainer
- Connection points and sinks

#### Asynchronous Download

- Downloading data—URLs
- Asynchronous monikers
- URL monikers

#### Controls, Embedding, and Categories

- Embeddable objects
- Properties/methods/events

#### **Control Persistence**

- Object persistence
- Streams and Internet Explorer
- Property bags
- Persistence in ATL
- ATL property maps

#### Windowless Controls

- ◆ Inactive controls—mouse interaction
- Container support for windowless controls
- Message dispatching

#### Code Download and Signing

- Internet component download and
- Packaging options

92

# **CGI Programming with Perl**

#### 5 days

#### Description

The Common Gateway Interface, or CGI, is a standard for accessing external programs through a Web server. It is commonly used in Web development for adding interactivity to Web pages and for linking to external data sources. CGI programs are most commonly developed using the Perl programming language. This module provides a solid introduction to both the Perl programming language and to writing CGI scripts using Perl.

#### Audience

Developers and system administrators who wish to develop programs or CGI scripts on either Unix or Windows 95/98/NT platforms using the Perl programming language.

#### Prerequisites

Programming experience is helpful, but not required. Basic knowledge of HTML is required. If the course is run on Unix, experience using Unix and Unix-based text editors is also required.

#### Hardware and Software

Perl 5 compiled and installed on the platform of choice. PCs or workstations capable of running a Perl 5 development environment. A CGI-capable Web server. The systems should be able to load a 3.5" DOS-formatted solution disk.

#### Topics

#### **Getting Started**

Perl Basics-Operators and Flow Control

Operators
 various operators
 operator precedence

#### Flow Control

#### Working with Scalar Variables in Perl

- Integer and Floating Point Scalars
- String Scalars

#### Working with List Variables in Perl

- Introduction to List Variables
- Working with Lists
- Manipulating Lists

#### Working with Hashes in Perl

- Introduction to Hashes
- Working with Hashes

#### Reading, Writing, and Manipulating Files in Perl

#### Pattern Matching in Perl

- Basic Pattern Matching
- Advanced Pattern Matching

#### Format

- Presentation
- 🖪 Hands-on

## Objectives

After completing this course, participants should be able to:

- Explain the history and structure of the Perl programming language
- Define CGI and explain its use in Web development
- Describe Perl syntax
- Write Perl programs
- Explain the process flow for CGI Web applications
- Write CGI scripts to handle HTML form processing
- Use Server Side Includes (SSI) for server-side processing
- Build complex CGI applications that require sessions, such as shopping carts

## **Coding Efficiently Using Subroutines**

#### CGI Basics

- CGI Basics
- Environment Variables
- Input/Output

#### Form Processing

- General Form processing
- Debugging CGI Programs
- ♦ CGI.pm

#### Form Fields, SSI, OO-Perl

- Detailed HTML Form Fields
- ♦ SSI, URL Rewriting
- Object Oriented Perl

#### Shopping Carts and Sessions

- Reading/Writing Files
- Sessions and Shopping Carts
- Cookies

#### Appendix A: Advanced Topics

- References
- Packages
- Modules

Internet – Web

#### Description

This intensive course provides a solid introduction to the Perl programming language. Hands-on exercises reinforce key concepts.

#### Audience

Developers and system administrators who wish to develop systems or application programs on either Unix or Windows platforms using the Perl programming language.

#### Prerequisites

Familiarity with C, procedural programming, or shell scripting is beneficial.

#### Hardware and Software

PCs or workstations capable of running a Perl development environment. The systems should be able to load a 3.5" DOS-formatted solution disk. Perl 5+ properly installed on the platform of choice.

#### Format

- Presentation
- 🖪 Hands-on

#### Objectives

After completing this course, participants should be able to:

- Write useful Perl scripts to solve common problems
- Use flow control, operators, lists, and hashes
- Understand and use pattern matching (regular expressions)
- Create subroutines and use library routines
- Create and use references
- Create and use modules
- Employ object-oriented programming techniques

# Internet — Web

#### Getting Started

Topics

- Running Perl programs
- Basic statement syntax
- Comments, variables, basic operators and functions
- Reading from standard input, writing to standard output and standard error

#### Operators

- Arithmetic, assignment, relational, logical, bitwise, and string operators
- Precedence and associativity

#### **Flow Control**

- If-else, elsif, and unless statements
- The warn(), die(), and exit() functions
- While and until loops

#### Working with Scalars

- Scalar variables; character strings
- Escape sequences

#### Working with Lists

- Defining and using list variables
- Assigning list elements to other variables
- Command line arguments
- Working with list ranges and list slices

#### Working with Hashes

- Creating hashes from lists
- Adding and deleting hash elements

#### **Reading and Writing to Files**

- Opening a file
- Determining file status
- Using pipes

#### Pattern Matching

Substitution and translation operators

#### **Creating and Using Subroutines**

- Formatting Output
- Value formats

#### File and Directory Functions

· File access, link, and directory manipulation functions

#### Process, Scalar, and List Functions

 Process, mathematical, string, scalar, list manipulation, and time functions

#### System Variables and Options

- Separator and pattern match variables
- -n and –p options

#### References

Creating multidimensional arrays

#### Packages and Modules

Importing modules into a program

#### **Object-Oriented Programming in Perl**

Creating and using classes

# Visual InterDev 6.0 and Active Server Pages

#### 3 days

#### Description

This course provides developers with a practical, hands-on introduction to Web site development with Microsoft's Visual InterDev. The course begins with an introduction to the Active Server Pages environment, including Internet Information Server (IIS), HTML, and VBScript. The course continues with coverage of Web database programming and developing Active Server Pages.

#### Audience

Intranet and Internet application designers and programmers who need a practical knowledge of developing Web sites using HTML, Active Server Pages, and VBScript.

#### Prerequisites

Familiarity with Visual Basic syntax, HTML programming, and database programming are recommended but not required.

#### Hardware and Software

Pentium-class PCs with at least 32MB RAM and 1GB disk. TCP/IP network (does not have to be connected to the Internet). Windows NT Server 4.0 with Service Pack 3 or later. Internet Information Server 3.0 or later, Visual InterDev 6.0 or later, Active Server Pages & FrontPage 98 Extensions, FrontPage 98 or later (optional), Internet Explorer 4.0 or later, Microsoft Access 97 or later.

#### Topics

#### Introduction

- Architecture of a dynamic Web site
- Client and server scripting
- Web page database access
- Visual InterDev development environment

#### Internet Information Server Overview

- IIS features
- · Installing and managing Web applications with IIS
- ♦ TCP/IP networks
- IIS management, logging, and security

#### **HTML Programming Overview**

- History of HTML
- HTML lists, links, tables, frames, and forms
- ♦ Cascading Style Sheets

#### **VB Scripting Overview**

- Client- and server-side scripting
- VB scripting syntax and object models

#### Format

- Presentation
- 🖪 Hands-on

#### Objectives

- After completing this course, participants should be able to:
- Design Web applications
- Develop Active Server Pages
- Utilize ActiveX Data Objects to develop Web database applications
- Work with design-time controls
- Embed Visual Basic scripting in Web pages
- Embed and program ActiveX controls in Web pages
- Define and describe IIS
- Deploy Web applications

#### ActiveX Controls

- Adding and programming ActiveX controls
- Design time controls

#### **Active Server Pages**

- Overview of Active Server Pages
- Using Server Objects
- Server components

#### Active Data Objects

- ◆ ADO Objects and Object model
- ADO transactions and examples

#### **Related Applications**

- Microsoft Index Server
- Reporting
- Microsoft Transaction Server
- Microsoft SMTP Server
- Microsoft NNTP Server
- Microsoft Exchange Server
- Microsoft Site Server
- Microsoft Visual Source Safe

# Introduction to XML

#### 2 days

#### Description

XML is an emerging technology that provides a language- and platformneutral means for creating text formats for structured data. This two-day course explains XML from a technical perspective, its role in multi-tier Web applications and provides hands on experience with XML and its companion technologies. The course covers all of the basics of XML: syntax, DTDs, Data Schemas, Namespaces, XSL, CSS, and the DOM API. During the course participants write a short XML document, write a DTD and Data Schema for their document, and format their document for display using CSS and XSL.

#### Audience

This course is intended for technical managers, Web developers or Web application developers who are, or will be, involved in developing and deploying XML solutions.

#### Prerequisites

No prior knowledge of XML is required. Familiarity with HTML and general programming concepts is beneficial.

#### Hardware and Software

Windows 95/98/ME/NT/2000 PCs, Internet Explorer 5+.

#### Format

- Presentation
- 🖴 Hands-on
- ♥ Interactive Activities

#### Objectives

After completing this course, participants should be able to:

- Define XML and its relationship to both HTML and SGML
- Contrast the roles of XML and HTML
- Identify the principal XML companion technologies and their uses
- Identify the features of well-formed XML
- Identify the features of valid XML
- Write a well-formed XML document
- Write a DTD and a Data-Schema
- Validate an XML document against a DTD and a Data Schema
- Use CSS to format an XML document for display on a Web browser
- Use XSL for format an XML document for display on a Web browser
- Use the DOM API to extract pieces of information from an XML document

#### Topics

#### **XML Basics**

- Metalanguages and Markup Languages Metalanguages: SGML, XML Markup Languages: HTML, XML-defined
- XML's Role in the Middle-Tier Communication between User Tier and Middle Tier Communication between Data Tiers and Middle Tier
- XML Components Part 1 Structure Overview: DTDs, XML Data Schemas Display Overview: CSS, XSL
- XML Components Part 2 Linking Overview: Xlink
- Parsing Overview: DOM, SAX

#### Well-Formed and Valid XML

- Well-formed XML
   syntax
   content
- Valid XML
   DTDs in detail
   Data Schemas in detail

#### **Displaying XML**

- ♦ CSS
- ◆ XSL XSLT
  - XSLT and HTML

#### Document Object Model (DOM)

- DOM Structure
- Principal DOM Interfaces Methods
  - Attributes
- DOM Scripts (using JavaScript)

96

Course # 02-7810
# XML Programming

### 4 days

### Description

The eXenstible Markup Language, or XML, is a technology that is rapidly gaining momentum for its flexibility and efficiency it brings to the development of Web-based solutions. Once touted as a replacement for HTML, XML is now being used for much more–from transaction processing to middleware to data exchange.

This 4-day hands-on course explains XML from a practical perspective: Web-based application development using XML. The course focuses on using XML with Active Server Pages, although using XML with Java is also briefly discussed. Topics include schema creation, XML parsing, document object model manipulation, and display rendering with XSL.

### Audience

This course is intended for Internet application developers who are, or will be, involved in developing and deploying XML solutions.

### Prerequisites

Students must be familiar with HTML, Dynamic HTML, and Cascading Style Sheets. Familiarity with JavaScript and Active Server Pages is extremely helpful. Familiarity with object-oriented concepts is beneficial.

### Hardware and Software

Windows NT 4+ (Workstation or Server), Internet Information Server 4, Internet Explorer 5, Microsoft Access (optional).

# Format

- Presentation
- 🖪 Hands-on
- Demonstration
- ♥ Interactive Activities

# Objectives

After completing this course, participants should be able to:

- Construct XML Document Type Definitions (DTDs) and XML Schemas, and describe valid and well-formed document syntax
- Avoid name conflicts with XML Namespaces
- Apply XLink and XPointer technology to extend XML document linking capabilities
- Manipulate parsed XML documents using JavaScript and the XML Document Object Model (DOM)
- Apply Cascading Style Sheets (CSS) and Extensible Stylesheet Language (XSL) to format and display XML documents
- Construct powerful and flexible multi-tier applications using XML technology

### Topics

# Introduction

- XML vs HTML
- XML uses and support
- XML and XSL examples
- Document structure
- Valid and well-formed documents

### **Document Structure**

- ♦ XML Document Type Descriptions
- XML-Data Schemas
- Advanced XML Topics
- Namespaces
- ♦ XLink and XPointer

### XML and Data Source Objects Overview

- ♦ XML Data Source Objects
- Applying data binding to XML

### JavaScript and ASP/JSP

- JavaScript
- HTML Document Object Model
- Active Server Pages

### XML DOM

- ♦ XML Document Object Model
- Using the XML DOM

### XML Presentation

- ◆ Formatting XML with HTML/CSS
- Formatting XML with XSL
- XML in 3-Tier Model
- XML in the data tier
- XML in the business tier
- XML and Java

Internet — Web

# XML Programming with Java

### 3 days

### Description

This course focuses on the use of the JAXP 1.1 (Java API for XML Processing) for parsing and transforming XML documents with a Java application. This course covers the parsing and validation of XML using both the SAX (Simple API for XML) and DOM (Document Object Model) parsers and the integration of these parsers into Java-based applications. Additionally, the course covers the use of XSLT to transform XML documents into multiple formats.

### Audience

The course is intended for Web and application developers who are interested in learning to use XML and Java together in order to leverage the platform independence and versatility of each.

### Prerequisites

The course assumes a working knowledge of both XML and Java: for example, completion of Arkoa's *Introduction to XML* and *Java2 Programming*.

### Hardware and Software

Each course participant needs a Windows 98, ME, NT, or 2000 PC, or a Unix Workstation with the following software:

- J2SE version 1.3 or above
- ♦ JAXP 1.1

Internet — Web

### Topics

# XML and Java Overview

- XML Overview Introduction Components of XML DOM and SAX Comparison
- JAXP Overview Introduction Overview of JAXP 1.1 JAXP Parsers Package JAXP Transform Package

# SAX: Simple API for XML

- Introduction to SAX Introduction SAX Parser Products JAXP 1.1 Parser Support SAX Version 2 Introduction to Using SAX Creating Factory and Parser Parsing Introduction to Handlers Configuration Example Code
- SAX Handlers Introduction DefaultHandler Class ContentHandler Interface

# Format

- Presentation
- 🖪 Hands-on
- Demonstration

# Objectives

After completing this course, participants should be able to:

- Describe the differences between SAX and DOM parsers
- Describe the JAXP 1.1 API and its support for parsing and validating XML documents
- Use SAX and DOM parsers together with Java to manipulate and validate XML documents.
- Use XPath, XSL, and XSLT to control styles.
- Write XSL stylesheets and use XSLT together with Java to transform an XML document into multiple formats

Other Useful Interfaces Handling Errors and Warnings

# Document Object Model (DOM)

- DOM API Overview Introduction DOM Parsers DOM Nodes Overview DOM Tree Example Introduction to Using DOM
  - Using JAXP to Support DOM
  - DOM Interfaces
  - DOM Error Handling
- XML Document Traversal and Modification Working with XML Documents Modifying a XML Document

### XSL, XPath, and XSLT

- Introduction to XPath *Xpath Overview XPath Expressions Xpath Functions XPath Abbreviations*
- XSLT Introduction *XSL Overview XSLT Overview Common XSLT Programming Constructs XSLT Processor*

XSLT—Templates and Adding Elements and Attributes
 Working with Templates
 Controlling Output

# Transforming XML Documents Using JAXP

• Obtaining and Configuring a Transformer

Introduction Using JAXP to Obtain a Transformer The Transformer Factory Handling Errors and Warnings URIResolver

Transforming XML

Introduction Using Transformer Using StreamSource or Result Using DOMSource or Result Using SAX Source or Result

# **Object Technology Technical Overview**

#### 1 day

### Description

This technical overview provides an introduction to the concepts and benefits of object-oriented techniques. The overview describes the concepts of encapsulation, abstract data types, message passing, inheritance and polymorphism. The differences between object-oriented design and traditional methods are discussed and the important organizational implications of adopting an object-oriented approach are considered.

### Audience

IT (information technology) managers and staff who are transitioning to an object-oriented software development approach.

### Prerequisites

An understanding of business applications and familiarity with the process of analysis and design for a typical software development project are helpful.

### Classroom Requirements

Overhead projector, flipcharts

### Topics

### Introduction to Object Technology

- Why object technology? Driving forces for object technology Computing context
- Object-oriented concepts Objects everywhere Object concepts and characteristics Classes and their characteristics

### **Object-Oriented Analysis**

- Analysis methods and notations Object-oriented development life cycle Incremental model Analysis and design methods
- Performing object-oriented analysis Review of analysis techniques Review of analysis notation
- Perform analysis

# Format

### Presentation

# Objectives

After completing this course, participants should be able to:

- Explain the driving forces for object technology and describe its evolution
- Define terms related to object technology and explain fundamental object-oriented concepts
- Define object-oriented analysis and design
- List and describe leading analysis and design methodologies
- Explain object-oriented analysis techniques and the process of objectoriented design
- List and describe tools and languages used for object-oriented development
- Define and describe components and component frameworks
- Describe the roles and skills needed for object-oriented development

### **OO Design and Development**

- Object-oriented design GUI design considerations Perform design
- Object-oriented tools
   Languages
   Class libraries
   Craphical user interfe
  - Graphical user interface tools Object-oriented databases Development considerations

### Implementing Object Technology

- Distributed component frameworks
   Distributed objects
   Object management architecture and the common object
   Compound document framework
   COM/DCOM and OLE
- Building a career in object technology Changing skill requirements in the development life cycle The project team's roles and responsibilities Skills and training

# Appendix: Object Technology Definitions

# **Object-Oriented Analysis and Design**

### 5 days

### Description

This intensive five-day course provides a practical working knowledge of object-oriented analysis and design using the Object Modeling Technique (OMT). The methodology is applied to a case study in order to see where and how concepts are best implemented. This course emphasizes effectiveness of using a consistent, robust methodology in object-oriented design through extensive written exercises.

### Audience

Systems analysts, designers and developers who will design and develop object-oriented programs.

### Prerequisites

Participants should have experience in analysis and design plus an overall understanding of object-oriented concepts.

# Format

- Presentation
- Written Exercises

# **Objectives**

٠

After completing this course, participants should be able to:

- Describe the object model and how to apply it
  - Derive an object model, dynamic model and functional model via an analytic process
- Create an object-oriented system design
- Recognize the current strengths and limitations of object-oriented analysis and object-oriented design

### Topics

### **Object-Oriented Concepts**

# • The need for object technology

### **Object Model**

- Class and object diagrams
- Attributes and operations
- Associations and links
- Multiplicity (cardinality)
- Aggregation and inheritance

### Dynamic Model

- States and events
- Specialization and concurrency ٠
- Advanced actions ٠

### Functional Model

- Transforms and terminal transforms ٠
- Data stores and control flows ٠
- Hierarchy of functions
- Definition of terminals

### **Relationship of OMT Models**

### **Object-Oriented Analysis**

- Analysis vs. design
- Constructing an object model
- When to build a dynamic model ٠
- When to use a functional model ٠
- ٠ Iteration

# System Design

- Subsystems
- Concurrency and processor allocation
- Control architecture
- Design trade-offs

### **Object Design**

- Combining the models
- Implementing control
- Maximize inheritance
- Object representation and storage

#### Pragmatics

- Immaturity of OOA/OOD
- New metrics
- Change and reuse ٠
- Development considerations

**Object Technology** 

# UML Object-Oriented Analysis and Design

### 5 days

### Description

This intensive five-day course provides a practical working knowledge of object-oriented analysis and design. Participants apply analysis and design processes using UML notation to a case study in order to see where and how concepts are best implemented. This course emphasizes the effectiveness of using a consistent, robust methodology in object-oriented design through extensive written exercises.

### Audience

Systems analysts, designers and developers who will design and develop object-oriented programs.

### Prerequisites

Participants should have experience in analysis and design plus an overall understanding of object-oriented concepts.

# Format Presentation

Written Exercises

# Objectives

After completing this course, participants should be able to:

- Describe the analysis and design process
- Utilize UML diagrams at different stages of the development life cycle
- Derive use cases and create use case diagrams
- Create static structure, behavior, and implementation diagrams via an analytic process
- Create an object-oriented system design

### Topics

### **Object-Oriented Concepts**

• Objects, classes, inheritance, polymorphism

# Introduction to Analysis and Design Using UML

- Building models
- What is UML?

### **Constructing Use Cases**

- Use cases and extend relationships
- Primary and secondary scenarios

# **Discovering Potential Classes Using CRC Cards**

What is CRC?

**Object Technology** 

### Responsibilities and collaborators

- Sequence Diagrams
- Diagramming behavior
- Sequence diagrams

# **Modeling Structural Elements**

- Methodology
- Operations
- Aggregation and composition
- Dependencies
- Constraints

### Modeling Dynamic Behavior

- States and events
- Notation for actions and activities
- Activity diagrams

### Packages

- Rendering packages
- Describing the hardware
- Patterns
- Frameworks

### The Unified Software Development Process

- The promise of OOADP
- Elaboration
- Construction

### The Requirements Capture Case Study

- Identifying actors and use cases
- Use cases

### The Analysis Process Case Study

• Brainstorming

### **Detail and Architectural Design Process**

- Discovering design abstractions
- Using patterns
- Mapping of databases
- Legacy data
- Designing components and interfaces
- Assessing risk
- Connectivity and communications
- Security
- Error handling and logging
- Moving into the construction phase

102

# Requirements Analysis and Modeling with Use Cases

### 2 days

## Description

This intensive, hands-on course addresses the requirements analysis process by applying Use Case modeling techniques. Through a series of lectures and lab exercises, course attendees will thoroughly explore designing and developing Use Cases and using them to gather and analyze requirements.

This course can be taught with or without a visual modeling tool such as Rational Rose or Visio although no specific training is provided on these tools.

### Audience

Business Analysts, Business Users, System/Software Analysts, Software Designers, and anyone else who is involved in requirements capture, specification, use, or management.

### Prerequisites

Participants should be familiar with the basics of analysis and design and the software development lifecycle.

### **Classroom Requirements**

Room set up as horseshoe and large enough for team activities, or with break-out rooms for team activities. Overhead projector, flip charts.

# Format

- Presentation
- Written Exercises
- Interactive Activities

### Objectives

After completing this course, participants should be able to:

- Describe the requirements analysis process, industry standards, and best practices
- Apply Use Case modeling to define business and system requirements
- Develop Use Case models
- Explain the components of a Use Case
- Explain how Use Cases drive design, development, testing, and documentation activities
- Apply Use Case modeling for change and scope management during the lifecycle of a project
- Describe the tools that can be used for Use Case modeling

# Topics

# **Requirements Analysis**

- The Requirements Analysis Process Industry Standards and Best Practices Unified Modeling Language (UML) Rational Unified Process (RUP) The Requirements Matrix
- Business Process Modeling Tools for Business Process Modeling Activity Diagrams

# **Use Cases and Actors**

- Use Case Diagrams
   Tools for Use Case Modeling
   Components of a Use Case Diagram
- Actors Types of Actors Relationship between Actors and Use Cases

# Components of a Use Case

- Introduction to Components
- Components
   Input and Output
   Frequency and Issues
- Super Use Case and Sub Use Case

# Design and Development with Use Cases

- Uses and Extends Relationships
- Applying Use Cases *Custom Development Package Development Use Cases as Test Scripts*
- Role of Use Cases Level of Detail Managing Scope Creep Change Management Traceability Check

### Use Case Modeling Activity

- Modeling Use Cases: Start to Finish Develop Activity Diagram Identify Actors Identify Use Cases
   Develop Use Cases
   Model Use Case-Actor relationship Model Uses relationship
   Model Extends relationship
   Develop Use Case Diagram
- Resources

Course # 12-0205

# Introduction to Rational Rose

### 1 day

### Description

This course provides a hands-on introduction to the Rational Rose modeling tool. The course reinforces object-oriented modeling practices and UML notation concepts as participants are exposed to the modeling and code generation capabilities of the Rose tool.

### Audience

System/Software Analysts, Software/Application Designers, and anyone else who is involved in object modeling, including Project, Technical, and QA Managers. This course is particularly targeted to those analysts and designers building object models for their systems.

### Prerequisites

Familiarity with object-oriented concepts is required. Some understanding of UML notation is recommended.

### Hardware and Software

Rational Rose 2000 installed on each system.

### Topics

### **Course Introduction**

- Course Objectives
- Approach
- Content

**Object Technology** 

104

- Reviewing Design Process
- Van Vliet Case Study

### Introduction to Rational Rose

- Choosing a Modeling Tool
- Rational Rose Versions
- Launching Rational Rose
- Rose Desktop
- Model Views
- Model Structure and Traceability
- Model Setup
- Model Options
- Specifying Project Documentation

#### Use Cases

- Use Case Diagrams
- Use Case Notation
- Customizing Notation Selection
- Identifying Actors
- Identifying Use Cases
- Use Case Documentation
- Use Case Enhancements
- Editing Model Components
- Customizing Look and Feel

### Activity Diagrams

- Use Case Scenarios
- Diagramming Scenario Activity Flow
- Activity Diagram Notation
- Defining Activities
- Pick-Up-Car Activity Diagram
- Activity Diagram Enhancements
- Designing Interface Prototypes

# Format

- Presentation
- 🖪 Hands-on

# Objectives

After completing this course, participants should be able to:

- Describe Rational Rose 2000, its features, and functionality
- Define and describe Rose conventions and terms
- Diagram use cases with Rose
- Discover and add classes to Rose
- Model class relationships
- Describe how to create sequence, collaboration, and state diagrams
- Generate and reverse engineer code

### Classes

- Identifying Classes
- Refined Class List
- Class Model Notation
- Adding Classes
- Associations
- Van Vliet Associations
- Van Vliet Class Diagram
- Class Attributes and Operations
- Associations or Attributes?
- Adding Operations

### **Dynamic Behavior**

- Analyzing Dynamic Behavior
- ♦ Sequence Diagram Notation
- Pick Up Car Sequence Diagram
- Collaboration Diagrams
- Collaboration Diagram Notation
- Specifying Object Links
- Pick-Up-Car Collaborations
- Completing the Class Model
- Enhanced Class Diagram
- State Diagram Concepts
- State Diagram Notation
- Defining State Transitions
- States and Superstates
- Vehicle State Diagram
- Van Vliet Subsystems
- Subsystem Design Notation

### Transition to Implementation

- Transition to Implementation
- Component Concepts
- Component Diagram Notation
- Component Assignment
- Van Vliet Components

- Reverse Data Engineering
- Forward Data Engineering
- DDL Generation
- Deployment Diagram
- Creating Controllable Units
- Rose Code Engineering
- Rose Reverse Code Engineering
- ♦ Java Code Generation
- Reverse Engineering Java
- ♦ C++ Code Generation
- Reverse Engineering with C++

# **Additional Features**

- Rose Web Modeler
- ♦ Van Vliet Web Site
- Rose Model Integration
- Model Integration
- Resolving Merge Errors
- Generating Project Documentation
- Rose Web Publisher
- Managing OO Development
- Rose SourceSafe Interface
- Using the Object Repository
- ♦ Rose -> Requisite Pro
- RequisitePro -> MS Project
- Rose Extensibility Language

# Advanced Object-Oriented Design Using Patterns and Frameworks

### 4 days

### Description

This intensive four-day course provides a practical working knowledge of object-oriented design with UML, patterns, and frameworks. Through written exercises, participants gain confidence in building effective designs with the UML notation and well-known design patterns and frameworks. In the application of UML, patterns, and frameworks to a case study, participants gain understanding in how to complete an object-oriented design. Course examples and exercises use Java.

### Audience

Systems designers and developers who will analyze, design, and develop object-oriented programs.

### Prerequisites

Attendance at Arkoa's 5-day UML Object-Oriented Analysis and Design course or equivalent training or work experience.

# Format

- Presentation
- 😐 Hands-on
- Demonstration
- Written Exercises

### Objectives

After completing this course, participants should be able to:

- Solve complex object based computing problems using patterns
- Identify specific problems that can be solved with design patterns
- Select the appropriate design pattern to solve a computing problem
- Document pattern-based computing solutions using UML diagrams
- Select the most appropriate solution to a computing problem between a pattern or a framework
- Build useful frameworks to create generic solutions to common OO design problems

### Topics

# Introduction

- N-tier architecture
- Introduction to design patterns

### Abstract Factory and Singleton Design Patterns

- Abstract factory design pattern
- Singleton design pattern

### **Creational and Composite Design Patterns**

- Builder pattern
- Factory method pattern
- Prototype pattern
- Composite design patterns

### Interpreter and Visitor Design Patterns

- Interpreter design pattern
- Visitor design pattern

### **Behavioral and Bridge Patterns**

- Behavioral patterns overview
- Command method pattern
- Iterator
- Mediator pattern
- State pattern
- Bridge design pattern

### Facade and Structural Design Patterns

- Façade design patterns
- Structural patterns
- Adapter pattern
- Decorator method pattern
- Flyweight pattern
- Proxy pattern

#### Java Application Framework

- An application framework
- Framework development
- Enterprise frameworks
- Enterprise JavaBeans architecture
- EJB and Java technologies
- Distributed objects
- ♦ RMI
- JDBC
- Servlets

# Introduction to Unix

### 2 days

### Description

This foundation course for Unix system administrators and programmers provides a general understanding of the Unix system and teaches the basic skills for using it. Although the course is based on the Korn shell, it is suitable for users of the C and Bourne shells.

### Audience

Systems and applications programmers and system administrators who plan to use the Unix operating system.

### Prerequisites

Basic computer literacy.

### Hardware and Software

Any Unix system, ideally with the Korn shell, but C or Bourne can be used. The systems should be able to load a 3.5" DOS-formatted solution disk.

# Format

- Presentation
- 🖪 Hands-on

# Objectives

After completing this course, participants should be able to:

- Describe the concepts underlying the Unix system
- Use a variety of basic Unix commands
- Describe the Unix tools approach enabling the use of most Unix commands
- Create, access, and manipulate data under Unix using the standard editors
- Recognize how the Unix file system works
- Describe how processes interact with each other and with files

# Topics

### Unix Overview

- Unix history and features
- Processes
- Software tools techniques

### Introductory Unix Commands

- Logging in
- Command syntax
- Command execution
- Documentation

# Introduction to Editing with vi

- vi as a text file viewer
- vi as a text file editor

# The Unix File System

- Files and directories
- File hierarchy
- Pathnames
- Directory entries
- File manipulation commands
- Links and symbolic links
- Protection
- Changing file protection: chmod
- Initial protection modes: umask
- Special files
- Multi-volume file systems

# The Korn Shell Command Interpreter

- The command interpreter
- Korn Shell user interface
- Filters and pipelines
- Filename generation and metacharacters
- Job control
- Command grouping and sub-shells
- More vi
- Advanced features of vi

### **Everyday Unix Tools**

• File manipulation and more

**Unix/Linux** 

# **Unix Fundamentals**

### 5 days

### Description

This course provides a basic introduction to the Unix system and skills for using it. Emphasis is placed on providing experience with the basic Unix facilities and commands in relation to data manipulation, shell programming, and text editors. Appreciation of the Unix tools philosophy is taught, enabling students to understand most Unix commands.

### Audience

Systems and applications programmers who plan to use the Unix operating system.

### Prerequisites

Participants must have experience programming in a high-level language. Some experience with other operating systems is advantageous but not essential.

### Hardware and Software

Any Unix system that provides the Korn shell. The systems should be able to load a 3.5" DOS-formatted solution disk.

### Topics

#### Unix Overview

• Unix history and features

### Introductory Unix Commands

- Command syntax and execution
- Processes
- Documentation

### Introduction to Editing with vi

- Using basic vi features
- vi modes
- Using advanced vi features
- Using colon commands
- Global commands
- File manipulation

### The Unix File System

- File system components
- Managing files

Unix/Linux

108

- Moving, renaming, and copying files
- Listing files and directory contents
- File system components
- File system hierarchy
- Working with directories
- Protecting files

# The Korn Shell Command Interpreter

- Using basic shell features
- Filters and pipelines
- Filename generation and metacharacters
- Escaping special characters
- Login start-up files
- Background processes
- Job control
- Aliases
- Command-line editing

# Format

- Presentation
- 🖪 Hands-on

# Objectives

After completing this course, participants should be able to:

- Use a variety of basic Unix commands
- Create, access, and manipulate data under Unix using the standard editors and other important Unix tools
- Program the command interpreter the Korn shell
- Recognize the interactive features of the Korn shell
- Describe how the Unix file system works

### More vi

Advanced features of vi

### **Everyday Unix Tools**

- File manipulation tools
- Getting information about files
- Comparing files
- Operating on file contents
- Miscellaneous tools
- Evaluating expressions
- Finding files
- Using electronic mail
- Combining tools

### Korn Shell Programming

- Writing and running shell scripts
- Interpretation and substitution
- Command substitution
- Shell variables and variable substitution
- Arithmetic with shell variables
- Positional parameters, shell parameters
- Blank interpretation and comments
- Getting user input read
- · For, while, and until loops, if statements
- Korn shell extended test facilities
- Conditionals
- Here documents
- Subshell type designator
- **General Purpose Tools**
- ♦ sed
- ♦ awk

### Software Development Tools

- ♦ ar
- SCCS
- make

# **Unix Tools and Utilities**

### 3 days

## Description

This is a foundation course for programmers. It provides an understanding of the Unix system and teaches the basic skills for using it. It does not focus on specific programming techniques or design methodologies, but places emphasis on providing hands-on experience with the basic Unix facilities and commands for data manipulation and shell programming.

### Audience

Systems and applications programmers who plan to use the Unix operating system.

### Prerequisites

Participants must have some experience with Unix, including the use of the vi editor and the simple filesystem. Arkoa's *Introduction to Unix* or *Introduction to Linux* are recommended prerequisites. Experience programming in a high-level language such as C is also useful.

### Hardware and Software

Any Unix system that provides the Korn shell. The systems should be able to load a 3.5" DOS-formatted solution disk.

# Topics

### The Korn Shell Command Interpreter

- The command interpreter
- Filters and pipelines
- Filename generation and metacharacters
- Shell variables, environment variables
- Controlling the login environment
- Job control
- Show process status ps
- Command grouping and subshells
- Aliases
- Command history
- Command-line editing

### **Everyday Unix Tools**

- Viewing file contents
- Getting information about files, comparing files
- Operating on file contents
- Simple text manipulation
- Traversing the file hierarchy find
- Miscellaneous commands
- Summary of most-useful commands

### Korn Shell Programming

- Writing and running shell scripts
- Interpretation and substitution
- Command substitution

# Format

- Presentation
- 🖪 Hands-on

# Objectives

- After completing this course, participants should be able to:
- Use many of the key Unix tools
- Describe the Unix tools approach to creating powerful user and programmer utilities
- Create, access, and manipulate data under Unix using important Unix tools
- Create Korn shell script programs
- Describe the Unix project archiving system

- Shell variables and variable substitution
- Shell environment variables
- Arithmetic with shell variables
- Positional parameters
- Shell parameters
- Blank interpretation
- Comments
- Getting user input read
- Modifying the command line shift
- The for, while and until loops and if statement
- Exit status or return code
- The test command
- Break and continue
- The case statement
- Simple conditionals
- The set command
- Here documents

### **General Purpose Tools**

- ♦ sed
- ♦ awk

### Software Development Tools

- ♦ ar
- SCCS
- make

**Unix/Linux** 

# **Unix System Administration**

### 3 days

### Description

This course is a practical introduction to the Unix operating system and the role of the system administrator. Students will gain enough knowledge and experience with the Unix V.4 system to enable them to provide system administration services.

### Audience

People who will be responsible for installing and administering Unix V.4 systems. The course is also appropriate for technical support engineers who will be providing Unix support.

### Prerequisites

Basic familiarity with Unix commands.

# Hardware and Software

A fully installed Unix V.4 system, with tape device, terminals, a printer and associated cabling. The sysadm menu system should be configured. The systems should be able to load a 3.5" DOS-formatted solution disk.

### Topics

#### Introduction to System V

- Administering a Unix system
- The filesystem reorganization

#### **Everyday Unix Tools**

- Viewing, comparing, and operating on files
- Traversing the file hierarchy find
- Summary of most useful commands

### Korn Shell Programming

• Korn shell programming constructs

#### Administration

- System administration tools
- Creating and removing a user
- User start-up
- Process control
- Communicating with users

### Using the Clock

The at and cron utilities

### The Filesystem

- The filesystem layout
- Partitions and device driver files
- Mounting and unmounting a filesystem
- Filesystem failure
- Fsck
- System owned files

## Format

- Presentation
- 🗕 Hands-on

# Objectives

- After completing this course, participants should be able to:
- Administer a Unix V.4 system
- Use all major system administration procedures
- Configure a Unix V.4 system
- Use filesystem management and maintenance
- Set up and use the security mechanisms available under Unix V.4
- Write basic shell scripts and modify existing ones
- Recognize problems and implement basic troubleshooting techniques

### Backups

- Methods of backup
- Incremental vs. full backup
- Backup tools

### **Booting and Shutting Down Unix**

- Unix system states
- Shutting down from multi-user and single-user modes
- Booting Unix
- Single-user mode
- ♦ init

#### **Terminal Management**

### Line Printer Management

- · Setting up a new printer
- Maintenance utilities
- Printer queues

# System Security

- SUID files
- File modes
- Passwords and checks

Unix/Linux

### 5 days

### Description

This course is a practical extension to *Unix System Administration*, covering the more complex administration requirements of networked and large Unix systems.

### Audience

Administrators of large Unix systems or networks of Unix systems. The course is also valuable for technical support staff for these systems.

### Prerequisites

Experience in Unix system administration.

### Hardware and Software

A Unix network running Unix V.4, with NFS, TCP/IP, and NIS all running. Terminals, printers, and associated cabling should be provided. Any Unix system V-based network with NFS, NIS, and TCP/IP may be substituted for the prior requirement.

### Topics

### **Unix History**

### Kernel and System Configuration

# **Processor Management**

- Booting and system shutdown
- Stand-alone utilities

## **File Systems**

- Filesystem structure
- Creating and mounting filesystems
- Filesystem reorganization
- Monitoring disk space

# Bad block handling

### **Kernel Overview**

- Process a detailed definition
- Traps and interrupts
- Memory management
- Process management
- Delayed read/write
- Inter-process communication

### Managing a Local Area Network

- An overview of the Internet Protocol Suite
- Useful network commands
- Network daemon processes
- Network configuration files
- Setting up a Local Area Network (LAN)
- Adding a node to the network
- Setting up multiple networks

# Format

- Presentation
- 🖪 Hands-on

# Objectives

After completing this course, participants should be able to:

- Install and configure a System V.4 environment
- Install new devices and software
- Set up networks and networked file systems
- Use the X Window system

# **Distributed Filesystems-NFS**

- RPC, XDR, and NFS
- The portmapper
- The virtual filesystem
- Exporting filesystems
- Mounting filesystems
- NFS mount options
- Nfsd and Biod processes
- File locking
- Troubleshooting tools

### The X Windows System

• X, Open Look, and Motif

### **Printer Administration**

- The SVR4 print spooler
- Troubleshooting

### **Unix Security Issues**

### **Networked Unix Security Issues**

- Network security problems
- UUCP and Berkeley TCP/IP security features
- NFS security features

### Performance Monitoring

- Performance monitoring tools
- Improving disk utilization
- Reducing prime-time load
- Kernel tuning
- · Real-time process scheduler control

# Introduction to Linux

### 2 days

### Description

This course provides a technical introduction to the major features and functionality of the Linux operating system.

### Audience

The target audience includes anyone new to Linux including users, developers, and system administrators.

### Prerequisites

General experience using personal computers is strongly recommended.

### Hardware and Software

A fully installed Linux system. The Linux examples given in the course are for Red Hat Linux version 6.0+. The systems should be able to load a 3.5" DOS-formatted solution disk.

# Format

- Presentation
- 😐 Hands-on

## Objectives

After completing this course, participants should be able to:

- Describe the major features of the Linux operating system
- Describe file and file system concepts
- Navigate the Linux file system and manage files
- Use the vi text editor
- Use the Bash shell command interpreter to execute commands
- Use common user-level Linux tools

### Topics

### Introducing Linux

- What is Linux?
- Using Linux
- Root and user accounts
- Command execution
- Using the online manual
- Some basic commands and tools

### The Linux File System

- File system components
- Managing files
- Moving, renaming, and copying files
- Listing files and directory contents
- File system components
- File system hierarchy
- Working with directories
- Protecting files

### Editing with vi

- Using basic vi features
- vi modes
- Using advanced vi features
- Using colon commands
- Global commands
- File manipulation

### The Shell Command Interpreter

- Using basic shell features
- Filters and pipelines
- Filename generation and metacharacters
- Escaping special characters
- Login start-up files
- Background processes
- Using advanced shell features
- Accidental file loss and logout protection
- ♦ Job control
- Aliases
- Command-line editing

### **Everyday Linux Tools**

- File manipulation tools
- Getting information about files
- Comparing files
- Operating on file contents
- Miscellaneous tools
- Evaluating expressions
- Finding files
- Using electronic mail
- Combining tools

Unix/Linux

# **Linux Tools and Utilities**

### 3 days

### Description

This course provides a technical introduction to common tools and utilities used with the Linux operating system, specifically Red Hat version 6.0.

### Audience

The target audience includes anyone interested in Linux tools and utilities including users, developers, and system administrators.

### Prerequisites

General experience using personal computers is strongly recommended. Some experience with Linux and/or Unix, equivalent to Arkoa's *Introduction to Unix* or *Introduction to Linux* course, is required.

### Hardware and Software

A fully installed Linux system. The Linux examples given in the course are for Red Hat Linux version 6.0. The systems should be able to load a 3.5" DOS-formatted solution disk.

# Format

- Presentation
- 😐 Hands-on

# Objectives

After completing this course, participants should be able to:

- List common Linux tools and utilities
- Set up configure, and use the X Window system
- Set up configure, and use the GNOME desktop
- Use floppy drives and CD-ROM drives in Linux
- Use file compression utilities
- Use the Emacs text editor
- Use ftp
- Use email, chat, and news utilities in Linux
- Use the Red Hat Package Manager
- Use gcc and make files to program in Linux
- Debug program code using gdb

# Topics

# The X Window System

- X fundamentals
  - The X Server and the X Client
    - Supported hardware X Server selection
    - Window managers
    - Desktop environments
- Configuring, starting, and using X *X configuration* 
  - Startup
  - X security or access control Remote display of X applications Common X applications
- Getting started with GNOME
   GNOME desktop components
   Using GNOME applications and utilities
   Customizing the GNOME interface

### **File Utilities**

- Using floppy disks and CD-ROMs
- File compression and packaging File compression utilities File packaging utility GNOME Midnight Commander file manager
- The Emacs editor
- Other file utilities
- gawk, ispell, Ghostscript and Ghostview

# Remote Execution and File Transfer Utilities

- Remote login and command execution
- File transfer utilities

### **Network User Communication Tools**

- Email clients in Linux
  - Elm, Pine, Mutt, Exmh, Netscape Messenger
- News readers in Linux *Tin, Netscape Newsgroup Reader*
- Talk and chat tools
- Web browsers in Linux Lynx, Netscape

### Using Red Hat Package Manager (RPM)

- RPM features and functionality
- Querying packages
- Installing and uninstalling Overview
- Miscellaneous RPM tools

### **Programming Utilities**

- Compiling C programs with gcc
- Building applications with make
- Introduction to debugging with gdb

# Linux System Administration

### 3 days

### Description

This course is a practical introduction to the Linux operating system and the role of the system administrator. Students will gain enough knowledge and experience with the Linux (Red Hat) system to enable them to provide basic system administration services.

### Audience

Linux system administrators or users interested in learning more about Linux system administration.

### Prerequisites

Some experience using the Linux or Unix operating system is strongly recommended. Participants should have completed Arkoa's *Introduction to Linux* course or have equivalent knowledge/experience.

### Hardware and Software

A fully installed Linux system and printer (optional). The Linux examples given in the course are for Red Hat Linux version 6.0. The systems should be able to load a 3.5" DOS-formatted solution disk.

### Topics

### Linux Administration Introduction

- Introduction to Linux
- Linux system administrator responsibilities

# **Everyday Linux Tools**

- File manipulation tools
- Commands
   Finding files
   Combining tools

### **Bash Shell Programming**

- Shell programming basics
- Flow control

### **User and Process Administration**

- User administration tasks Account administration
- Process administration tasks *Prioritizing processes Cron jobs*

# Format

- Presentation
- 🗕 Hands-on

### Objectives

After completing this course, participants should be able to:

- Describe the major responsibilities of a Linux system administrator
- Use common user-level Linux tools
- Program shell scripts using the Bash shell
- Administer users and processes
- Administer the file system
- Perform backups
- Control boot and shutdown of the Linux OS
- Administer printers and terminals
- Setup basic system security

### The File System

- File system concepts
   Partitions and device driver files
   Swapping
- File system administration
- Mounting, creating and checking

#### Backups

- Backup concepts
- Backup tools
  - tar, cpio, and dd

### **Booting and Shutting Down Linux**

- Boot and shutdown concepts
- System initialization
   System initialization file
   RC scripts

### **Basic Peripheral Administration**

- Terminal administration
- Printer administration

### System Security

- Superuser security
- Physical and login security

<u>Unix/Linux</u>

# Advanced Unix Tools

### 3 days

### Description

This course teaches students how to utilize the full range of advanced Unix tools. It instructs students in the use of Unix tools to perform complex programming tasks and in programming the Bourne shell.

### Audience

Systems and applications programmers, project leaders and technical managers who are using the Unix operating system to develop sophisticated programs or who are responsible for large Unix installations.

### Prerequisites

Participants must have a good grounding in the use of Unix. Familiarity with the C programming language is desirable.

### Hardware and Software

Any Unix system with the Bourne shell available.

# Topics

# **Regular Expressions**

# Tools for Tree Traversal

- tar tape archiver
- ♦ find
- ♦ cpio

# Simple Text Manipulation Tools

- ♦ basename
- ♦ dirname
- ♦ tr
- ♦ expr

# **Tools For Finding Patterns**

- ♦ grep
- Sorting Tools
- ♦ sort
- ♦ uniq

# Advanced Shell Programming

- Redirection
- Shell variables and parameters
- Flow of control
- Advanced techniques: debugging, evaluation, exec, exit, and set

# **File Comparison Utilities**

Diff, sdiff, comm, cmp

- Presentation
- Hands-on

# **Objectives**

After completing this course, participants should be able to:

- Solve complex problems using the tools that Unix provides
- Assess the suitability of the various Unix tools for performing sub-tasks
- Apply shell programming techniques to solve problems efficiently
- Improve productivity by mastering the Unix environment
- Describe the facilities for controlling source code

# Chopping and Changing

Split, csplit

# cut, paste and join

- Job Control Tools
- Background jobs
- Scheduling considerations
- Hang up
- Cron tables
- Job control: fg and bg, jobs, kill and wait

# More KSH Features

- Pattern matching
- Alias
- Arithmetic variables
- Environment variables
- Arrays
- Awk- A Supertool
- awk and commands

# Editing

sed

# Software Development Tools

- ♦ ar
- ٠ sccs - source code control system
- make

# Format

115



# Advanced Programming in a Unix Environment

### 5 days

### Description

This course addresses POSIX standard 1003.1B, the System V inter-process communication package, the BSD sockets interface (a de facto standard for computer communications) and the Transport Layer Interface (TLI) , widely adopted as a standard for Open Systems communications.

### Audience

Systems and applications programmers working in a Unix operating system environment.

#### Prerequisites

Participants should have a good grounding in the Unix operating system. Since many of the exercises involve writing short programs in the C language, fluency in C is essential.

#### Hardware and Software

A Unix system, ANSI C compiler and debugger, POSIX 1003.1B compatibility, System V IPC, and BSD socket libraries. The systems should be able to load a 3.5" DOS-formatted solution disk.

# Format

- Presentation
- 🖪 Hands-on

# Objectives

After completing this course, participants should be able to:

- Write programs which interface with the Unix kernel
- Describe the Unix system call interface as defined in the IEEE standard POSIX 1003.1B
- Use various system calls
- Use the full range of facilities for inter-process communication in an appropriate manner
- Implement the client/server model across a network, using sockets on top of either TCP/IP or UDP/IP

### Topics

### Kernel Overview

- The waiter model
- Process a detailed definition
- Attracting attention traps and interrupts
- Resource allocation memory management
- Process management
- The filesystem
- A closer look at I/O
- Delayed read/write
- Inter-process communication

### System Calls

- Include files
- Process management
- Contextual information
- Files and directories
- Input and output
- Terminal handling
- Compliance with ANSI C
- System databases

### Inter-process Communication (System V)

- The client/server model
- ♦ FIFOs
  - Message queues
  - Shared memory
  - Semaphores
  - Record locking

#### Sockets

- Rationale behind sockets
- Fundamental concepts
- Implementation details
- Exceptions

Appendix: Transport Layer Interface (TLI)

Appendix: The Symbolic Debugger (SDB)

Appendix: The BSD Symbolic Debugger (DBX)

**Unix/Linux** 

# Windows NT Architecture Technical Overview

#### 1 day

### Description

This is a high-level, practical overview of the Windows NT architecture. This one-day overview describes each of the important Windows NT subsystems and explains how they interoperate. The course also provides an overview of the Windows NT filesystem, BackOffice Server, and Microsoft's Distributed intraNet Architecture (DNA). The major features of Windows 2000 are explained.

### Audience

IT professionals, including project managers, team or group leaders, analysts, designers, programmers, and support personnel.

### Prerequisites

Familiarity with computing systems and programming concepts.

### **Classroom Requirements**

Overhead projector and flipcharts.

### Topics

### Introduction to Windows NT Architecture

- Design goals
- Comparison of editions
- System architecture components
- Tools to view NT internals

### Windows NT Executive Subsystems

- An object management overview
- A security system overview
- Process management overview
- An LPC facility overview
- A memory management overview
- Win32 GUI subsystem overview
- An I/O system overview
- Cache manager overview

# Windows NTFS and File Server

- NTFS major features
- Comparison with FAT
- NTFS structure
- The Windows NT filesystem tree
- Windows NT file server

### Windows DNA

- A Windows DNA overview
- Windows DNA technologies

Presentation

# Objectives

Format

After completing this course, participants should be able to:

- Describe the functionality and components of Windows NT system architecture
- Describe the structure and functionality of Windows NT Executive components

Course # 11-0990

- Describe the structure and operation of Windows NTFS
- Describe the functionality and integration of BackOffice Server products with Windows NT
- Describe the structure and functionality of Windows DNA
- Describe the major new functionality and components of Windows 2000

### **BackOffice Server 4.0**

- ◆ A BackOffice products overview
- Windows NT option pack
- Other BackOffice products
- Deployment: Internet Bookstore
- Deployment: Satellite Offices

### Windows 2000

- Windows 2000 products
- Selected major features

### 5 days

### Description

This course focuses on design and development skills for the Microsoft Windows 95 environment. It emphasizes programming of consistent user interfaces in forms-based and multi-window applications. Exercises reinforce programming interface and concepts.

### Audience

Windows NT developers seeking to acquire the skills necessary to utilize the Win32 API.

### Prerequisites

C programming experience or attendance at Arkoa's *C Programming* course is required.

### Hardware and Software

Windows 95/98/ME/NT needs to be installed, with a 32-bit compiler suitable for that platform. The systems should be able to load a 3.5" solution disk.

### Topics

### Windows Concepts

- Windows system architecture
- Programming concepts
- System management

### **Building Windows 32 Applications**

- Build steps and settings
- Make files

### **Basic Windows Programming**

- Program layout
- The window procedure
- Registering classes
- Creating windows

### Messages

- Message concepts
- Message loop and common messages

### Unicode

- Internationalization
- Unicode and the Win32 API

# Painting and Text Output

- Display contexts
- WM\_PAINT processing
- Message Boxes

# Format

- Presentation
- 🖪 Hands-on

### Objectives

After completing this course, participants should be able to:

- Design and build a complete Windows 95 application
- Describe the concepts behind and implications of developing userdriven applications
- Use and program Windows 95
- Design standard user interfaces
- Use Windows 95 tools

### Menus and Accelerators

- Menu and accelerator concepts
- Design guidelines

# Command processing

- The Button Class
- Button styles
- Buttons as child windows

#### Resources

- Icons and cursor
- Loading resources

### Model Dialog Boxes

- Dialog box types
- Dialog procedure

The Edit Class

- Windows File I/O
- The Listbox Class
- Common Dialogs
- Modeless Dialog Boxes
- Drag and Drop

# Memory Management

# Processes and Threads

♦ DLLs

#### Course # 08-0100

# Windows NT API Programming

### 5 days

### Description

This practical, hands-on course focuses on design and development skills for the Microsoft Windows NT environment. Participants should be able to progress rapidly towards producing effective applications. Exercises reinforce programming skills and concepts.

### Audience

Windows NT developers seeking to acquire the skills necessary to utilize the Win32 API.

### Prerequisites

C programming experience or attendance at Arkoa's *C Programming* course.

### Hardware and Software

Windows NT with a 32-bit compiler suitable for that platform. The systems should be able to load a 3.5" solution disk.

# Topics

### Windows Concepts

- Windows system architecture
- Programming concepts
- System management

### **Building Applications**

- Build steps and settings
- Make files

# **Basic Windows Programming**

- Program layout
- The window procedure
- Registering classes
- Creating windows

### Messages

- Message concepts
- Message loop and common messages

### Unicode

- Internationalization
- Unicode and the Win32 API

# Painting Text

- Display contexts
- WM\_PAINT processing
- Message Boxes

# Format

- Presentation
- 🖪 Hands-on

# Objectives

- After completing this course, participants should be able to:
- Design and build a complete Windows NT application
- Describe the concepts behind and implications of developing userdriven applications
- Use and program Windows NT
- Design standard user interfaces
- Use Windows NT tools

# Menus and Accelerators

- Menu and accelerator concepts
- Design guidelines
- Command processing

# Buttons

- Button styles
- Buttons as child windows

# Resources

- Icons, stringtable, and cursor
- Loading resources

# Modal Dialogs

- Dialog box types
- Dialog procedure
- Edit Class

File I/O

- Common Dialogs
- Drag and Drop
- Memory Management
- Processes and Threads
- DLLs

Course # 08-0200

### 5 days

### Description

This practical, hands-on course focuses on design and development skills for the Microsoft Windows environment. A large set of the Win32 and Win64 Application Programming Interface is explored. This course is appropriate for those performing system-level programming for Windows 95/98/ME/2000.

# Audience

Windows developers seeking to build client applications, server components, and Windows services with the Win32/Win64 API, ATL, and ADSI.

### Prerequisites

Attendees should have C programming experience or attendance at Arkoa's *C Programming* course. C++ programming experience or attendance at Arkoa's *C++ Programming* course is also helpful.

### Hardware and Software

A PC for each student loaded with Windows 95/98/ME/2000 and a C/C++ compiler suitable for that platform.

### Topics

### Windows Concepts

- Windows system architecture
- Windows Distributed iNternet Architecture
- Programming concepts

### **Building Applications**

- Build steps and settings
- Make files

### **Basic Windows Graphical Programming**

- Program layout
- The window procedure
- Registering classes
- Creating windows
- Processing messages and message loops
- Basic GDI functions and the display context

### Unicode and the Win32 API

#### **Graphical Interfaces Elements**

- Menus and accelerators
- Controls and resources
- Dialogs

### Structured Exception Handling

- Termination and exception handlers
- Interoperability of Win32 SEH and C++ exception handling

### File I/O

General file operations and asynchronous I/O

# Format

- Presentation
- 🖪 Hands-on

### Objectives

After completing this course, participants should be able to:

- Design and build a basic Windows graphical application
- Take advantage of COM and COM+
- Understand Windows security
- Work with virtual memory and the heap
- Create and manage processes and threads
- Design and build basic Windows services
- Utilize files and pipes for input and output
- Manage the Active Directory with ADSI

### Memory Management

### Processes, Threads, and Synchronization

### Anonymous and Named Pipes

- Dynamic Link Libraries
- Creating DLLs
- Loading and using DLLs

# Windows Security

- Windows security overview
- Security Structures
- Impersonating a client

### Writing Windows 2000 Services

- Managing services
- Programming services
- The event log

### COM

- COM architecture and threading models
- Custom COM objects

#### **COM+ Services**

- Designing and developing transactional components
- Queued components

#### **Active Directory**

- Introduction to Active Directory
- ADSI COM-based API

120

# **Client/Server Visual Basic Development**

### 5 days

### Description

Visual Basic is widely regarded as the most popular Windows application development environment. Microsoft has continually enhanced Visual Basic's capabilities to include enterprise-wide distributed application development. This course explains how to develop and distribute Visual Basic applications in a client/server environment.

# Audience

Programmers and developers using Visual Basic 6.0 to develop client/ server applications.

### Prerequisites

Participants must have an understanding of Windows and basic programming skills. Familiarity with Basic syntax is helpful, but not necessary.

### Hardware and Software

Windows PCs for each student and instructor. Visual Basic 6.0 is required.

### Topics

# Introduction to Visual Basic

- Overview of Visual Basic
- Creating a Visual Basic application
- Visual Basic Forms
- Visual Basic forms
- Introduction to MDI

# **ActiveX Controls**

- Introduction to ActiveX controls
- Common controls

# Visual Basic Coding

- Coding basics
- Control statements and procedures
- Debugging

# Menus and User Input

- Menus
- Controlling user input

# **Dialog Boxes**

- Designing and creating dialog boxes
- Working with Windows: Files and the Registry
- ♦ File I/O
- The Windows Registry

# Format

- Presentation
- Hands-on

# **Objectives**

After completing this course, participants should be able to:

- Utilize and customize the Visual Basic IDE
- Describe the Visual Basic toolset
- Apply sound user-interface design
- Design and develop Visual Basic applications
- Enhance applications with OLE automation, distributed database access, and error handling
- Utilize external files from Visual Basic applications
- Describe and use ActiveX technologies
- Package and distribute completed applications

# **Exception Handling**

- Introduction to error handling
- Traps
- The Err object
- Raising errors

# Working with Common Controls

 ActiveX common controls, ImageList, ToolBar, StatusBar, SSTab, Rich-TextBox

# Working with More Common Controls

Animation, MSChart, UpDown, MSFlexGrid

# **Database Connectivity**

- VB database connectivity
- The ADO data control

# Database Programming with Active Data Objects (ADO)

- ADO Command and Recordset objects
- ADO Supporting topics

# Messaging with MAPI

MAPI messaging controls

# **OLE Integration and ActiveX**

- Application integration with OLE
- Introduction to ActiveX

# Distributing Visual Basic Applications

Application packaging

### 5 days

### Description

This course prepares programmers and application designers to develop advanced, enterprise-wide, distributed applications using Visual Basic, Active Data Objects, ActiveX technologies, and Microsoft Transaction Server.

### Audience

Application programmers and designers who need to design and develop distributed enterprise applications with Visual Basic.

### Prerequisites

Completion of Arkoa's *Client/Server Visual Basic Development* course, or an equivalent level of knowledge. Familiarity with object-oriented concepts and the Windows NT environment and architecture.

### Hardware and Software

Windows NT 4.0+ (Workstation or Server) with DCOM configured. Microsoft Transaction Server 2.0+. Microsoft SQL Server 6.5+. (Other ODBC compliant database servers can be accommodated). Visual Basic 6.0. Visual Studio is recommended (Visual C++ 6.0, Visual J++ 6.0, etc.), but optional.

### Topics

### **Client/Server Design Principles**

- Client/server design methodologies
- Tiered systems

### Database Programming with ADO

- Connection, command, and recordset objects
- Parameters, properties, and errors collection

### **Multiuser Database Considerations**

- Transactions
- Pessimistic and optimistic locking

### Internal Reuse

- Classes and objects
- Using the multiple interfaces

### **Object Model Implementation**

- Properties and methods
- Errors

### **ActiveX Components**

OLE, DLL, and EXE

# **ActiveX Components Design**

- Component design principles
- Compatibility

# Format

- Presentation
- 🖪 Hands-on

# Objectives

After completing this course, participants should be able to:

- Design and program multi-tier distributed applications with Visual Basic, ADO, and MTS
- Design clients and servers utilizing distributed ActiveX components
- Design and implement MTS components that control distributed database transactions

### **Distributed Components**

Tools and method blocking

### **ActiveX Controls**

### Advanced ActiveX Controls

- Binding controls
- VB and the Internet
- ActiveX documents
- **MTS Overview**

### **Developing an MTS Component**

- Stateful versus stateless components
- Installing MTS Components

# **Debugging MTS Components**

**Component-based Transactions** 

# MTS Security

Distributing MTS client applications

Windows Development

# Introduction to Visual Studio

### 1 day

### Description

This course provides an overview of Microsoft's Visual Studio, an Integrated Development Environment (IDE) for developing applications in Visual Basic, Visual C++, Visual J++, Visual InterDev, and Visual FoxPro. The course covers Visual Studio concepts such as project creation, building, and debugging. The course focuses on Visual Basic and Visual C++, using the same example application developed in both languages to demonstrate Visual Studio features.

# Audience

New or inexperienced programmers who are investigating or plan to use Visual Studio for a programming project.

### Prerequisites

Students should be familiar with Microsoft Windows user interface conventions and understand concepts such as toolbars and menus.

### Hardware and Software

Visual Studio version 6.0 properly installed on student PCs.

# Topics

# Visual Studio Basics

- Visual Studio Application Development Suite Microsoft DNS and DNA; Microsoft's COM/DCOM technology; Windows database technologies; enterprise application and multi-tier architecture
- Visual Studio environment Requirements; installation; Visual Studio IDE – Visual Basic, Visual C++; controls and components; Project Wizards

# Visual Studio Development

- GUI projects
  - Visual development; workspaces and projects; Visual Basic projects; Visual C++ projects
- Objects and controls
   Object-oriented programming; adding controls to projects and forms; event handlers
- Basic coding

### Code editors; data environments

# **Building Visual Studio Projects**

- Building and executing projects Application files; libraries; building Visual Basic and Visual C++ projects; execution
- Debugging projects

# Format

- Presentation
- 🖪 Hands-on

# Objectives

After completing this course, participants should be able to:

- Recognize and use the various components in the Visual Studio Interface
- Develop and debug Visual Basic and Visual C++ applications using Visual Studio
- Analyze and deploy projects built with Visual Studio
- Manage and use the Visual Source Safe revision control system for multi-user projects

# Management, Deployment, and Analysis

- Custom components
  - Creating and testing a custom component; adding the custom component to a project
- Deployment, analysis, and management Packaging; analysis and management

# Source Control System Overview

- File versus project based source control
- Visual Source Safe concepts
- Installation
- VSS Administrator, VSS Explorer
- Integrating VSS with Visual Basic and Visual C++
- Creating projects and adding files
- Get Latest Version, Check In, Check Out, Undo Check Out
- Differences

Show History; Show Differences; Journal files

- Sharing and branching
- Share files and pin files; branch files and merge branched files

### 2 days

### Description

This course provides an in-depth understanding of message queuing and its implementation using Microsoft Message Queue in creating multi-tier client/server applications.

### Audience

This course is intended for systems application developers and system administrators who are, or will be, involved in developing and deploying Microsoft Message Queue applications.

### Prerequisites

Completion of Arkoa's *Enterprise Development with Visual Basic and ActiveX* course, or an equivalent level of knowledge. Students must be conversant in Visual Basic (version 5 or greater) and Microsoft Transaction Server. Familiarity with object-oriented concepts and the Windows NT environment and architecture is beneficial.

### Hardware and Software

Windows NT 4.0+ (Workstation or Server) with DCOM configured. Microsoft Message Queue 1.0+, Microsoft Transaction Server 2.0+. Microsoft SQL Server 6.5+. (Other ODBC compliant database servers can be accommodated). Visual Basic 6.0. Visual Studio is recommended (Visual C++ 6.0, Visual J++ 6.0, etc.), but optional.

# Format

- Presentation
- 🗕 Hands-on

# Objectives

After completing this course, participants should be able to:

- Describe message queuing concepts
- Identify MSMQ architectural components
- Use the MSMQ Explorer to manage and monitor the MSMQ Enterprise
- Use the MSMQ object model and VB to create a queuing application
- Troubleshoot MSMQ problems
- Secure the MSMQ Enterprise
- Optimize design and system configuration for best performance

# Topics

# Introduction

### Understanding MSMQ

- Architecture, servers, clients
- Storing and routing messages
- MSMQ tools and samples
- MSMQ Explorer

# Managing and Monitoring MSMQ

- Navigating the MSMQ hierarchy
- Views
- Managing MSMQ

# Working with MSMQ in VB

- MSMQ ActiveX objects
- Working with queues
- Sending/receiving messages

# Troubleshooting MSMQ

- Viewing MSMQ error codes
- Installation errors
- Troubleshooting
- Error trapping

# MSMQ Case Study Exercise

# MSMQ Auditing and Security

- Auditing overview
- Security requirements and limitations
- Access control
- Authentication
- Encryption
- Securing MSMQ files and folders

# Optimizing MSMQ

- Single machine optimizations
- Recoverable mode considerations
- Multiple sender optimization
- Transaction coordination alternatives
- Authentication and encryption considerations
- Message header considerations
- Registry settings

Windows Development